

软件测试基础与实践

Software Testing: Foundations and Practices

第3讲 黑盒测试

教师：汪鹏 廖力

软件工程专业 主干课程



廖 力

本讲内容

黑盒测试
基本概念

黑盒测试定义、特点、意义、...

黑盒测试方法

等价类划分、边界值分析、因果法、
决策表、基于图/状态的测试、...

黑盒测试工具

主流黑盒测试工具



廖 力

— 黑盒测试概念

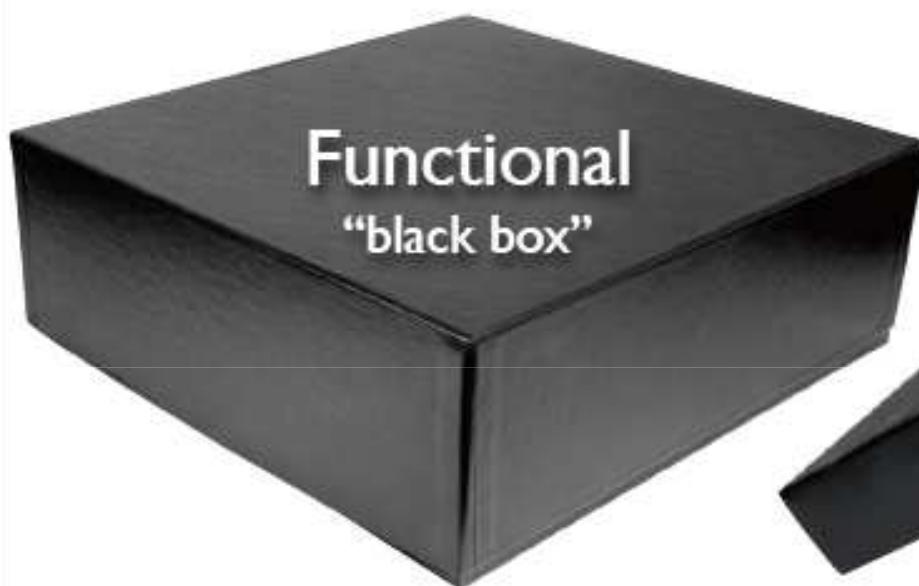


1. 为什么需要黑盒测试？
2. 黑盒测试有什么特点？
3. 如何实施黑盒测试活动？



廖 力

Testing Tactics



Functional
“black box”



Structural
“white box”

- Tests based on spec
- Test covers as much *specified* behavior as possible

- Tests based on code
- Test covers as much *implemented* behavior as possible

黑盒测试基本概念

■ 为什么需要黑盒测试

1. 有时无法获取程序代码
2. 尽早进行黑盒测试可以尽早发现软件功能缺陷
3. 弥补遗漏的逻辑缺陷
4. 适用于各个测试阶段：单元测试，集成测试，系统测试，回归测试



廖 力

黑盒测试基本概念

■ 定义

黑盒测试：一种**基于规格说明，不要求考察代码，以用户视角**进行的测试

其它称谓：**功能测试、基于规格说明的测试**

■ 意义

黑盒测试有助于软件产品的总体功能验证：

1. 检查**明确需求和隐含需求**
2. 采用**有效输入和无效输入**
3. 包含**用户视角**



廖 力

黑盒测试基本概念

■ 实施者

专门的软件测试部门；有经验的测试人员

■ 步骤

- (1) 规格说明书；
- (2) 生成测试用例；
- (3) 执行测试；
- (4) 判定测试结果

黑盒测试以一种系统化的方式依据规格说明生成测试用例，以避免出现冗余并提高覆盖率



廖 力

■ 黑盒测试面临的挑战：穷举？



用尽可能少的测试用例发现尽可能多的缺陷。



廖 力

■ 黑盒测试面临的挑战：编码逻辑未知 [Next](#)

问题：代码不可见前提下，面对未知的缺陷，如何设计测试用例？

解决1：随机测试？

解决2：在适当的方法指导下，**基于需求规格说明书**设计测试用例，既要有**正面测试**，也要有**负面测试**。
需要更强的技术和耐心来精心设计。

例如：黑客正是通过黑盒测试发现安全缺陷



廖 力

■ 黑盒测试随机进行是否可行? [Back](#)

随机测试

1. 均匀选取可能的输入值
2. 避免选择偏见
3. 对所有输入一视同仁

随机测试缺点

1. 程序缺陷分配并不均匀
2. 随机测试很难取到特定值



廖 力

黑盒测试方法基础 —基于需求的测试



廖力

东南大学 软件学院 College of Software Engineering

2019年10月21日

黑盒测试方法基础——基于需求的测试

■ 目的

确认软件需求规格说明书列出的需求

■ 前提

需求规格已经过仔细评审

隐含需求明确化



廖 力

黑盒测试方法——基于需求的测试

■ 需求规格说明样本

锁和钥匙系统需求规格说明

序号	需求标识	需求描述	优先级
1	BR-01	插入号码为123的钥匙并顺时针转动，应能上锁	高
2	BR-02	插入号码为123的钥匙并逆时针转动，应能开锁	高
3	BR-03	只有号码为123的钥匙可以用来上锁和开锁	高
4	BR-04	其它东西都不能用来上锁	中
5	BR-05	其它东西都不能用来开锁	中
6	BR-06	锁受重物撞击也不能被打开	中
7	BR-07	锁和钥匙重量必须为150g左右	低
8	BR-08	开锁和上锁转动方向可变，便于左手人士使用	低



廖 力

黑盒测试方法——基于需求的测试

■ 需求跟踪矩阵 (RTM)

需求跟踪矩阵样本

需求标识	需求描述	优先级	测试条件	用例标识	测试阶段
BR-01	插入号码为123的钥匙并顺时针转动，应能上锁	高	使用号码为123的钥匙	LOCK_001	单元组件
BR-02	插入号码为123的钥匙并逆时针转动，应能开锁	高	使用号码为123的钥匙	LOCK_002	单元组件
BR-04	其它东西都不能用来上锁	中	使用IC卡 使用发卡	LOCK_005 LOCK_006	集成
BR-06	锁受重物撞击也不能被打开	中	使用石头砸锁	LOCK_011	系统
BR-08	开锁和上锁转动方向可变，便于左手人士使用	低			未实现



廖 力

黑盒测试方法——基于需求的测试

■ 需求跟踪矩阵 (RTM)

作用：

1. 可跟踪每个需求的测试状态而不会遗漏任何需求
2. 优先执行优先级高的测试用例，尽早发现高优先级
区域内缺陷
3. 可导出特定需求对应的测试用例清单
4. 评估测试工作量和测试进度的重要数据



廖力

黑盒测试方法——基于需求的测试

■ 需求跟踪矩阵 (RTM)

需求跟踪矩阵传达的信息

测试指标	测试数据采集指标
1. 按优先级分类的需求数	1. 已通过的测试用例总数
2. 按需求分类的测试用例数	2. 未通过的测试用例总数
3. 已设计的测试用例数	3. 需求中的缺陷总数 4. 已完成的需求数 5. 挂起的需求数



廖力

黑盒测试方法——基于需求的测试

■ 需求跟踪矩阵 (RTM)

测试执行数据样本

序号	需求标识	优先级	测试用例	用例总数	通过用例	未通过用例	通过率	缺陷数
1	BR-01	高	Lock_001	1	1	0	100	1
2	BR-02	高	Lock_002	1	1	0	100	1
3	BR-03	高	Lock_003,004	2	1	1	50	3
4	BR-04	中	Lock_005,006,007	3	2	1	67	5
...
合计	8			12	10	2	83	12



廖力

黑盒测试方法——基于需求的测试

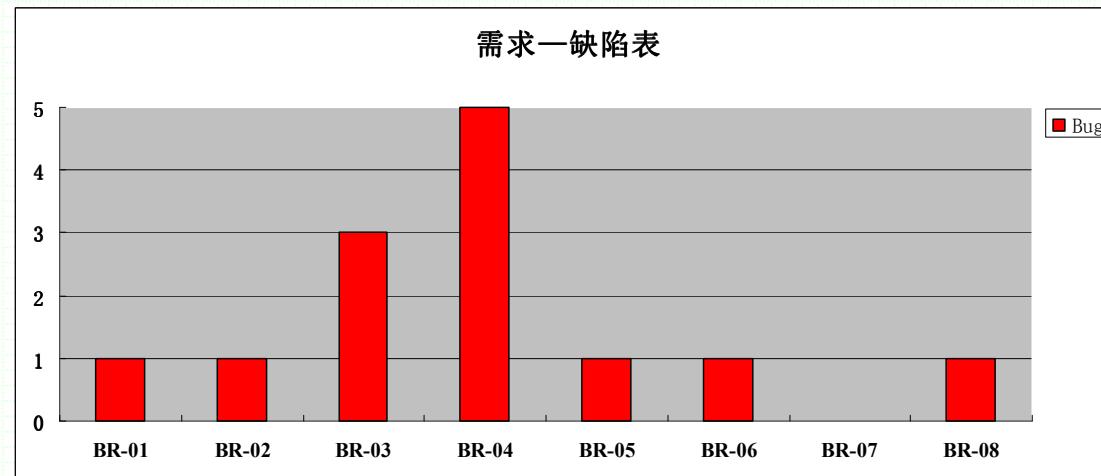
■ 需求跟踪矩阵 (RTM)

数据分析

1. 测试用例已通过率83%; 已满足需求71%;
2. 有1个高优先级需求BR-03没有通过测试, 对应3个缺陷;
3. 有1个中优先级需求BR-06没有通过测试, 对应5个缺陷;

.....

形成图表:



廖力

黑盒测试方法基础 —正面和负面测试



廖力

东南大学 软件学院 College of Software Engineering

2019年10月21日

黑盒测试方法基础——正面和负面测试用例

■ 正面测试

正面测试用例：

测试用例通过一组预期输出验证产品需求
目的：

证明软件对于每条规格说明和期望都能通过

注意：预期产品给出一个错误时它确实给出该错误，这也是正面测试的一部分

正面测试用于验证已知测试条件，
证明软件可以完成所期望的工作



廖 力

黑盒测试方法基础——正面和负面测试用例

■ 正面测试

正面测试用例举例

需求标识	输入1	输入2	当前状态	预期状态
BR-01	号码为123的钥匙	顺时针转动	开锁	上锁
BR-01	号码为123的钥匙	顺时针转动	上锁	不变
BR-02	号码为123的钥匙	逆时针转动	开锁	不变
BR-02	号码为123的钥匙	逆时针转动	上锁	开锁
BR-04	发卡	顺时针转动	上锁	不变



廖力

黑盒测试方法基础——正面和负面测试用例

■ 负面测试

负面测试：

展示当输入非预期输入时产品没有失败

目的：

使用产品没有设计和预想到的场景，尝试使系统垮掉

负面测试用于通过未知条件把产品搞垮



廖 力

黑盒测试方法基础——正面和负面测试用例

■ 负面测试

负面测试用例举例

序号	输入1	输入2	当前状态	预期状态
1	某个其它锁的钥匙	顺时针转动	上锁	上锁
2	某个其它锁的钥匙	逆时针转动	开锁	开锁
3	铁丝	逆时针转动	开锁	开锁
4	用石头打击		上锁	上锁

注意：负面测试用例不能映射到需求



廖 力

黑盒测试方法基础——正面和负面测试用例

■ 正面测试VS负面测试

	正面测试	负面测试
测试条件	根据需求说明规格	测试条件都在需求规格说明之外
目的	验证需求规格说明的正确性	通过未知条件搞垮软件
覆盖率计算	覆盖需求和条件	没有覆盖率
挑战性	根据规格说明设计测试用例	需要高度创造性产生尽可能多的未知条件



廖 力

二 黑盒测试方法



1. 如何设计黑盒测试的用例？
2. 各种黑盒测试方法的应用场景？
3. 如何结合不同的黑盒测试方法？



廖力

黑盒测试方法——概览

- 1. 等价类划分
- 2. 边界值分析
- 3. 因果分析法
- 4. 决策表
- 5. 基于图/状态的测试
- 6. 正交数组测试
-



廖 力

黑 盒 测 试 方 法 — 等价类划分



廖 力

东南大学 软件学院 College of Software Engineering

2019年10月21日

黑盒测试方法——等价划分

■ 等价划分方法基础

原理：

- 将程序的输入域划分为数据类，以便导出测试用例；
- 它试图定义一个测试用例以发现各类错误，从而减少测试用例数目，降低测试工作量。

等价类（划分）：

- 如果软件行为对一组值来说是相同的，则称这组值为等价类；
- 产生同一个预期输出的一组输入值叫一个划分



廖 力

黑盒测试方法——等价划分

■ 等价划分方法基础

有效等价类和无效等价类：

- 有效等价类：完全满足产品规格说明的输入数据，即有效的、有意义的输入数据构成的集合
- 利用有效等价类可以检验程序是否满足规格说明书
- 无效等价类：不满足程序输入要求或者无效的输入数据构成的集合

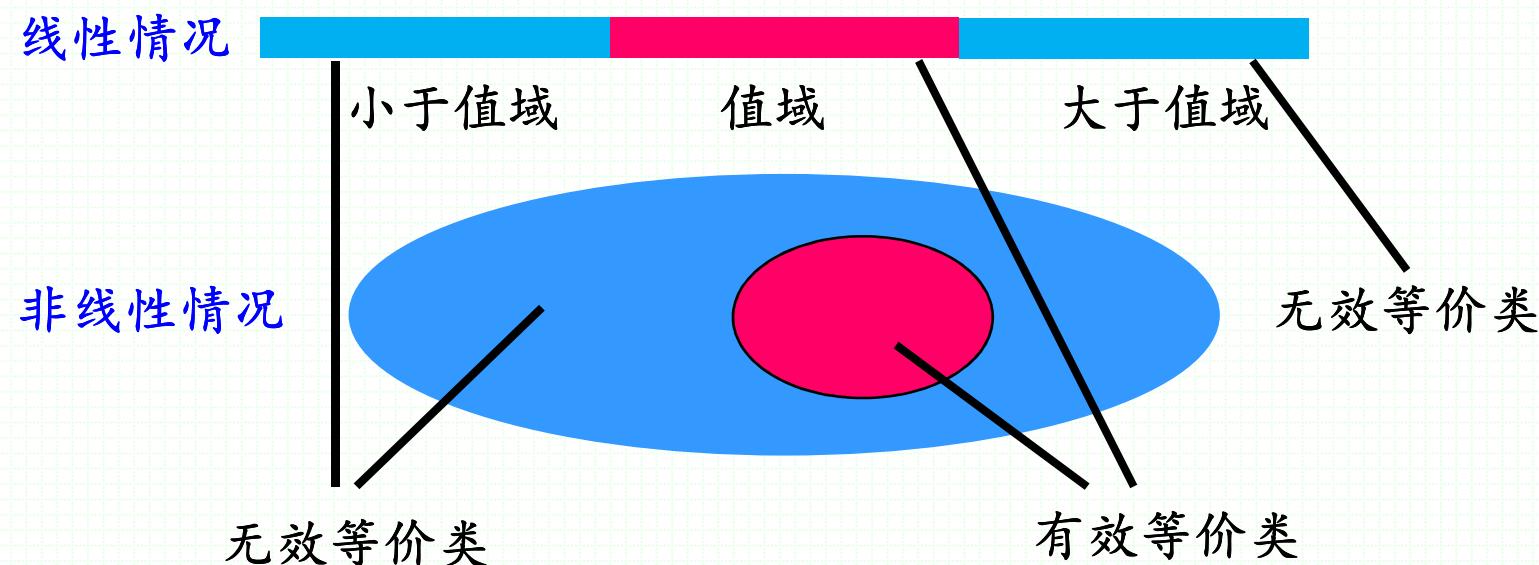


廖 力

黑盒测试方法——等价划分

■ 等价划分方法基础

有效等价类和无效等价类



廖 力

黑盒测试方法——等价划分

■ 等价划分准则

(1) 输入条件是布尔表达式，则可以定义一个有效等价类和一个无效等价类

例如：



Office 自动
更正选项

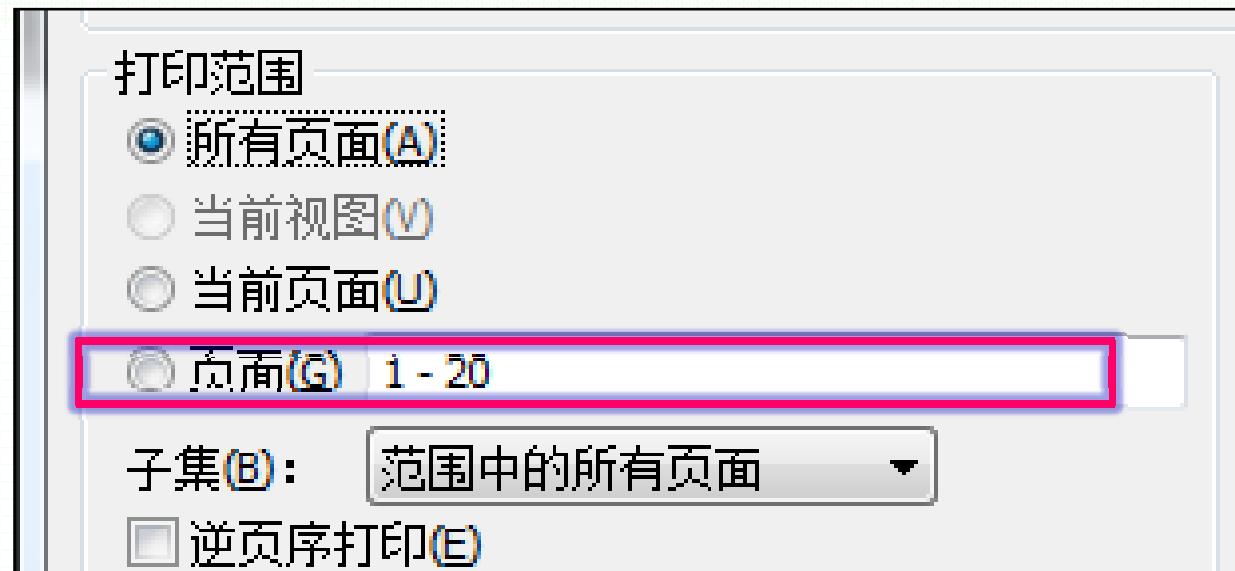


黑盒测试方法——等价划分

■ 等价划分准则

(2) 输入条件代表一个范围，则可以定义一个有效等价类和两个无效等价类；

例如：



廖 力

黑盒测试方法——等价划分

■ 等价划分准则

(3) 输入数据个数有规定，则可以定义一个有效等价类和两个无效等价类；

例如，规定输入构成三角形的3条边

有效等价类： 输入边数 = 3

无效等价类： 输入边数 < 3 和 输入边数 > 3



廖 力

黑盒测试方法——等价划分

■ 等价划分准则

(4) 输入条件代表集合的某个子集，则可以定义一个有效等价类和一个或多个无效等价类；

例如，邮政编码必须由6位数字构成

有效等价类：6位数字

无效等价类：6位字母 >6位字母 <6位字母

>6位数字 <6位数字

6位字母+数字...



廖 力

黑盒测试方法——等价划分

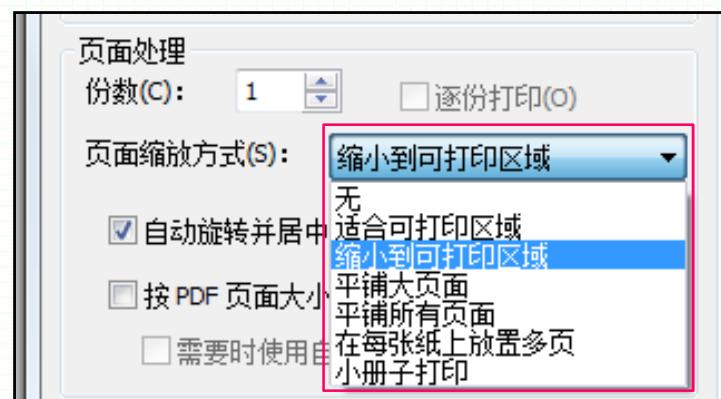
■ 等价划分准则

(5) 输入条件代表一组列表形式的数据，则可以定义N个有效等价类和一个无效等价类；

例如，中国的直辖市

有效等价类：北京、上海、天津、重庆

无效等价类：非直辖市



黑盒测试方法——等价划分

■ 等价划分准则

(6) 输入条件代表要求符合某几个规则，则可以定义多个有效等价类和若干个无效等价类；

例如，电子邮件地址规则：

必须含有@

@后格式为x.y

地址中不能包含某些符号，如/#&

有效等价类：[字母|数字]1..n@x.y

无效等价类：不含@ @后形式不是x.y 包含逗号



廖力

黑盒测试方法——等价划分

■ 等价划分准则

输入数据类型	划分等价类规则	
布尔值	1个有效等价类: TRUE	1个无效等价类: FALSE
取值范围	1个有效等价类: 正确取值范围	2个无效等价类: 大于和小于取值范围
数据个数	1个有效等价类: 正确数据个数	2个无效等价类: 大于和小于数据个数
集合	1个有效等价类: 正确的集合取值	1个或多个无效等价类
需分别处理的一组输入数据	多个有效等价类: 每个输入数据为1个等价类	1个无效等价类
符合某些规则的输入	多个有效等价类: 符合某个规则的输入数据为1个等价类	若干个无效等价类

力



黑盒测试方法——等价划分

■ 等价划分准则

实际情况往往是千变万化的，在遇到具体问题时，可参照上述规则的思想来划分等价类。



廖力

黑盒测试方法——等价划分

■ 等价划分方法基础

等价划分方法步骤：

Step1: 选择划分准则（范围、取值、布尔、集合…）

Step2: 根据准则确定有效等价类和无效等价类

Step3: 从等价类中选取样本数据

Step4: 根据需求写预期结果

Step5: 加入特殊值

Step6: 执行测试



廖力

黑盒测试方法——等价划分

■ 等价划分举例（范围）

人寿保险费率（基本月保险费0.50）

年龄段	额外保险费
35岁以下	1.65
35-59	2.87
60岁以上	6.00



廖力

黑盒测试方法——等价划分

■ 等价划分举例

人寿保险费率对应等价类

序号	等价类	输入类型	测试数据	预期结果
1	低于35岁	有效	26, 12	月保险费=2.15
2	35-59岁	有效	37	月保险费=3.37
3	大于60岁	有效	65, 90	月保险费=6.50
4	负年岁	无效	-23	警告信息



廖 力

黑盒测试方法——等价划分

■ 等价划分练习：注册用户名

合法的用户名要求：

- (1) 由字母开头
- (2) 后跟字母或数字的任意组合构成
- (3) 有效字符数不超过6个。



廖 力

字母开头

后跟字母或
数字

不超过6个
字符数

有效等价类

无效等价类

黑盒测试方法——等价划分

■ 等价划分练习：注册用户名

1个有效等价类：

(1) 用户名：字母+{字母|数字}0..5

4个无效等价类：

(2) 非字母开头的字符串集合

101, #1108

(3) 字母开头的字符串，并含有特殊字符

user@\$， user_01

(4) 字母开头的字符串，且长度超过6

user0001

(5) 空字符串



廖 力

软件测试

SOUTHEAST UNIVERSITY

Software Testing

序号	测试用例		覆盖
	输入	预期输出	
1	abc	合法	1
2	101	非法	2
3	user@\$	非法	3
4	user0001	非法	4
5		非法	5



廖 力

东南大学 软件学院 College of Software Engineering

2019年10月21日

黑盒测试方法——等价划分

■ 等价划分练习：注册用户名

3个有效等价类：

- (1) 用户名由字母开头
- (2) 后跟字母或数字的任意组合构成
- (3) 有效字符数不超过6个。



廖 力

黑盒测试方法——等价划分

■ 等价划分练习：注册用户名

4个无效等价类：

(4) 非字母开头的字符串集合

101, #1108

(5) 字母开头的字符串，并含有特殊字符

user@\$， user_01

(6) 字母开头的字符串，且长度超过6

user0001

(7) 空字符串



廖 力

软件测试

SOUTHEAST UNIVERSITY

Software Testing

序号	测试用例		覆盖
	输入	预期输出	
1	abc	合法	1, 2,3
2	101	非法	4
3	user@\$	非法	5
4	user0001	非法	6
5		非法	7



廖 力

东南大学 软件学院 College of Software Engineering

2019年10月21日

黑盒测试方法——等价划分

■ 等价划分练习：注册用户名

2个有效等价类：

- (1) 用户名由字母构成，字符数不超过6个
- (2) 用户名由字母开头，后跟字母和数字的任意组合构成，字符数不超过6个。



廖 力

黑盒测试方法——等价划分

■ 等价划分练习：电话号码

我国电话号码一般由两部分构成：

地区码：以0开头的3位或4位数字

电话号码：以非0、非1开头的7位或8位数字



廖 力

黑盒测试方法——等价划分

■ 等价划分练习：电话号码

输入数据	有效等价类	无效等价类
地区码	<ol style="list-style-type: none">1. 以0开头的3位区码2. 以0开头的4位区码	<ol style="list-style-type: none">3. 以0开头的小于3位的数字串4. 以0开头的大于4位的数字串5. 以非0开头 的数字串6. 以0开头的含有非数字的字符串



廖 力

黑盒测试方法——等价划分

■ 等价划分练习：电话号码

输入数据	有效等价类	无效等价类
电话号码	7. 以非0、非1开头的7位号码 8. 以非0、非1开头的8位号码	9. 以0开头的数字串 10. 以1开头的数字串 11. 以非0、非1开头的小于7位数字串 12. 以非0、非1开头的大于8位数字串 13. 以非0、非1开头的含有非法字符7或者8位字符串



廖力

黑盒测试方法——等价划分

■ 等价划分练习：电话号码

输入数据	有效等价类	无效等价类		
测试用例	010 6123456 025 81234567 0551 7123456 0571 92345678	覆盖1, 7 覆盖1, 8 覆盖2, 7 覆盖2, 8	01 81234567 05511 6123456 10 81234567 025x 81234567 010 06123456 0551 1123456 0551 612345 0517 912345678 0571 912345ab	覆盖3 覆盖4 覆盖5 覆盖6 覆盖9 覆盖10 覆盖11 覆盖12 覆盖13



廖力

黑 盒 测 试 方 法 — 边 界 值 分 析



廖 力

东南大学 软件学院 College of Software Engineering

2019年10月21日

黑盒测试方法——边界值分析

■ 边界值分析原理

软件的两个主要缺陷源：

- (1) 条件； (2) 边界

条件：变量取值需要采取的特定行动

边界：各种变量值的“极限”

边界值分析：

能有效捕获出现在边界处的缺陷的一种测试方法；
利用并扩展了缺陷更容易出现在边界处的概念



廖 力

黑盒测试方法——边界值分析

■ 边界值分析原理

缺陷出现在边界处的原因：

- (1) 使用比较操作符时未仔细分析
- (2) 多种循环和条件检查方法引起的困惑
- (3) 对边界附近需求的理解不够



廖 力

黑盒测试方法——边界值分析

■ 边界值分析原理

测试边界：

测试临近边界的有效数据，测试最后一个可能有效的数据，同时测试刚超过边界的无效数据

第1个减1/最后1个加1

开始减1/完成加1

最小值减1/最大值加1



廖 力

黑盒测试方法——边界值分析

■ 边界值分析原理

边界条件类型：

数据类型：

数值 速度 字符 地点 位置 尺寸 数量

特征：

第一个/最后一个 最小值/最大值

开始/完成 超过/在内

空/满 最短/最长 最慢/最快 最早/最迟

最大/最小



廖 力

黑盒测试方法——边界值分析

一些特殊的边界值

报表的第一行、最后一行 或 第一列、最后一列

循环的开始、结束

屏幕光标/鼠标的左上角、右下角

Int, long, unsigned int 整数的最小，最大值

e.g. [singned] long int 4位 -2147483648~2147483647

unsiged long [int] 4位 0~4294967295



廖 力

黑盒测试方法——边界值分析

■ 如何界定边界值

边界值附近数据的确认举例

	边界值附近数据	测试用例设计思路
字符	Min-1, Min, Min+1 Max-1, Max, Max+1	若文本输入框允许输入1-255个字符。 则边界值测试用例： 0,1,2,254,255,256
数值范围	Min-1, Min, Min+1 Max-1, Max, Max+1	输入数据允许 1-999 。 0, 1, 2, 998, 999, 1000
集合、空间	0, Min, Min+1 Max-1, Max, Max+1	图片大小不能超过 50k 0, 1k, 49k, 50k, 51k



廖 力

黑盒测试方法——边界值分析

■ 如何界定边界值

边界值附近数据的确认方法

n: 存在边界值的参数个数

m: 边界值条件数

Paul Jorgensen公式：

1. 4n+1: 基本边界测试

2. 6n+1: 健壮性边界测试

3. 3m: 边界条件测试

3种方法的测试粒度依次增强



廖 力

黑盒测试方法——边界值分析

■ 如何界定边界值

基本边界测试

Paul Jorgensen公式

1. $4n+1$: n为存在边界值的参数个数

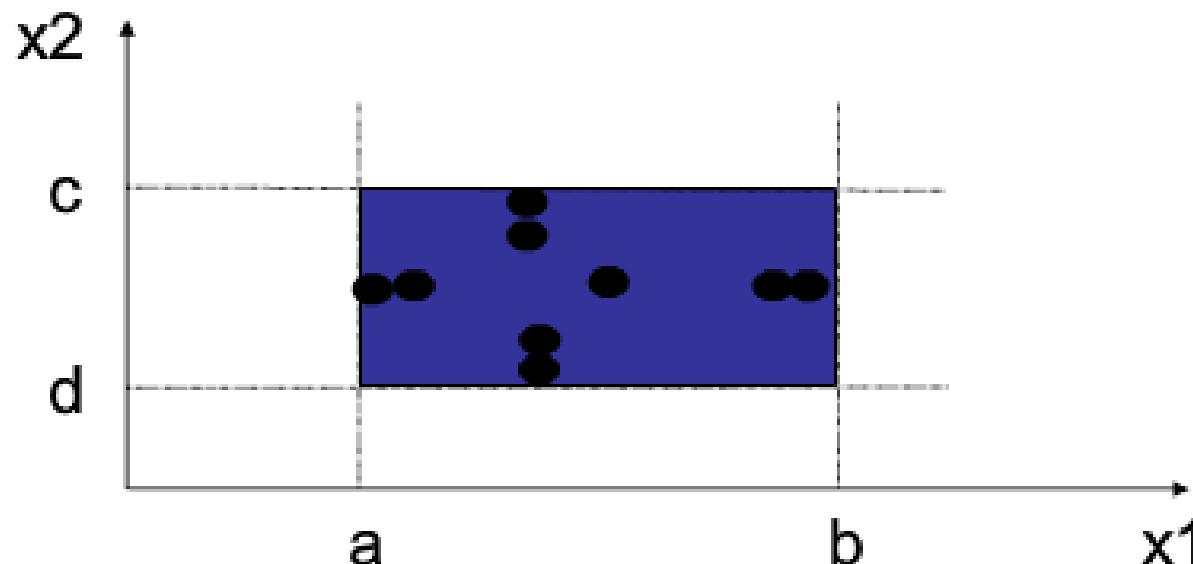
每个参数取**min, min+1, max-1, max**各一次，
同时其他参数取典型值。最后全部参数取典型值一
次。



廖 力

- 例1：有两个输入变量 $x_1(a \leq x_1 \leq b)$ 和 $x_2(c \leq x_2 \leq d)$ 的程序F的边界值分析
测试用例如下：

```
{ <x1nom,x2min>, <x1nom,x2min+>, <x1nom,x2nom>,
  <x1nom,x2max>, <x1nom,x2max->, <x1min,x2nom>,
  <x1min+,x2nom>, <x1max,x2nom>, <x1max-,x2nom> }
```



黑盒测试方法——边界值分析

■ 如何界定边界值

健壮性边界测试

Paul Jorgensen公式

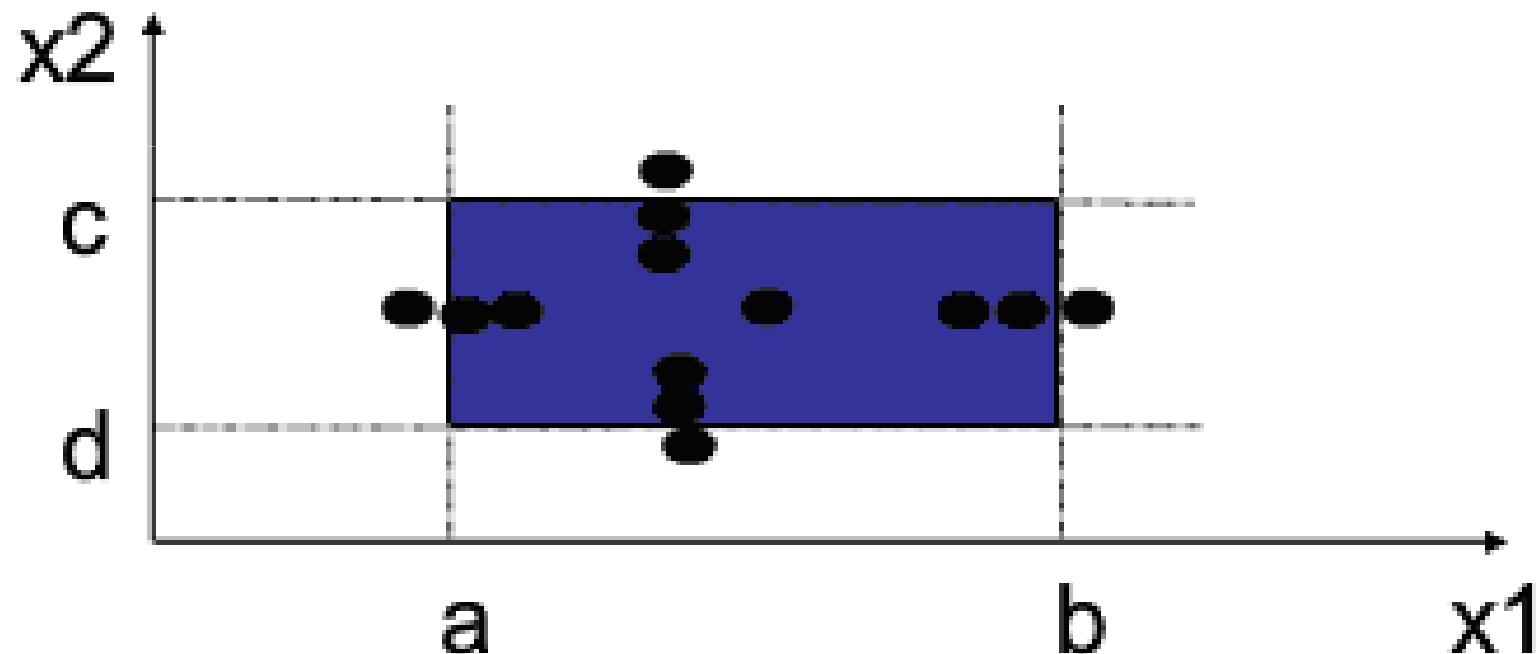
2. $6n+1$: n为存在边界值的参数个数

每个参数取**min-1, min, min+1, max-1, max, max+1**各一次，同时其他参数取典型值。最后全部参数取典型值一次。



廖 力

前面例1中的测试用例按 $6n+1$ 公式扩充如下：



黑盒测试方法——边界值分析

■ 如何界定边界值



问题描述:

Next Date程序需要输入3个整数，代表特定的月、日和年。同时，程序以月/日/年的格式返回下一日期。

Special notes:

计算1582年至3000年的日期



廖力

1. 参数: 健壮性边界测试

年 月 日 n=3

2. 确定边界值:

年: 1582, 3000

月: 1, 12

日, 1, 31

3. 测试用例设计:

方法二 ($6n+1$)

18+x个测试用例

测试用例	月	日	年	预期输出
1	0	1-27	1582-3000	错误
2	1	1-27	1582-3000	下一日期
3	2	1-27	1582-3000	下一日期
4	11	1-27	1582-3000	下一日期
5	12	1-27	1582-3000	下一日期
6	13	1-27	1582-3000	错误
7	1-12	0	1582-3000	错误
8	1-12	1	1582-3000	下一日期
9	1-12	2	1582-3000	下一日期
10	1-12	30	1582-3000	下一日期
11	1-12	31	1582-3000	下一日期
12	1-12	32	1582-3000	错误

测试用例	月	日	年	预期输出
13	1-12	1-27	1581	错误
14	1-12	1-27	1582	下一日期
15	1-12	1-27	1583	下一日期
16	1-12	1-27	2999	下一日期
17	1-12	1-27	3000	下一日期
18	1-12	1-27	3001	错误
19	1-12	1-27	1582-3000	下一日期

黑盒测试方法——边界值分析

■ 如何界定边界值

条件边界测试

Paul Jorgensen公式

1. $3m$: m 为边界值条件数

每个条件取-1, 自身, +1各一次。



廖 力

黑盒测试方法——边界值分析

■ 测试用例设计举例



大宗购买折扣

购买数量	单价
头10件 (1-10件)	5.00
第二个10件 (11-20件)	4.75
第三个10件 (21-30件)	4.50
超过30件	4.00

15件需要支付: $10 \times 5 + 5 \times 4.75 = 73.75$



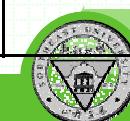
廖 力

软件测试

SOUTHEAST UNIVERSITY

Software Testing

测试用例	测试的输入值	选择理由	预期输出
1	0	第1阶段开始之前	0或报错
2	1	第1阶段开始	5.00
3	2	第1阶段开始之后	10.00
4	9	正好在第1阶段的末尾, 2阶段开始前	45.00
5, 7	10	极限值	50.00
6, 8	11	第2阶段开始	54.75
9	12	第2阶段开始后	59.50
10	19	正好在第2阶段的末尾, 3阶段开始前	92.75
11,13	20	极限值	97.50
12,14	21	第3阶段开始	102.00
15	22	第3阶段开始之后	106.50
16	29	正好在第3阶段的末尾, 4阶段开始之前	138.00
17	30	极限值	...
18	31	极限值+1	



廖 力

黑盒测试方法——边界值分析

■ 如何界定边界值



问题描述:

Next Date程序需要输入3个整数，代表特定的月、日和年。同时，程序以月/日/年的格式返回下一日期。

Special notes:

计算1582年至3000年的日期



廖 力

1. 参数: 条件边界测试

月 日 年 n=3

2. 测试用例设计:

方法三 (3m)

3m个测试用例

3. 确定边界条件:

(1) 每次只考虑一个参数的边界, 固定其它参数

(2) 补充确定的关联边界值

3.1 固定日、年的月边界条件

月: 1, 12

日: 1-27 年: 1582-3000

边界条件	月	日	年
1	1	1-27	1582-3000
2	12	1-27	1582-3000

3.2 固定月、年的日边界条件

日: 1, 30, 31

月: 31/30天的月

年: 1582-3000

边界条件	月	日	年
3	30天的月	1	1582-3000
4	30天的月	30	1582-3000
5	31天的月	1	1582-3000
6	31天的月	31	1582-3000

3.3 固定月、日的年边界条件

年：1582, 3000, 闰年，非闰年

月：1-12

日：1-27, 28, 29

边界条件	月	日	年
7	1-12	1-27	1582
8	1-12	1-27	3000
9	2	1	闰年
10	2	29	闰年
11	2	1	非闰年
12	2	28	非闰年

3.4 补充确定的关联边界条件

边界条件	月	日	年
13	1	1	1582
14	12	31	3000

$m=14$

总的测试用例数： $14 \times 3 = 42$

测试用例	边界条件	月	日	年	预期输出
1	1	0	1-27	1582-3000	错误
2		1	1-27	1582-3000	下一日期
3		2	1-27	1582-3000	下一日期
4	2	11	1-27	1582-3000	下一日期
5		12	1-27	1582-3000	下一日期
6		13	1-27	1582-3000	错误
7	3	31天月	0	1582-3000	错误
8		31天月	1	1582-3000	下一日期
9		31天月	2	1582-3000	下一日期
...

边界值分析测试练习——公历日期

用边界值分析方法设计合法公历日期的测试用例

日期范围：1582年1月1日—3000年12月31日

- ◆ 1582年10月5日至10月14日排除在公历外
- ◆ 2038年1月19日是BIOS提供的记时基准时间
1970年1月1日的最大值(千年虫问题)
- ◆ 英国1752年才采用阳历，他们扣除9/3/1752到
9/13/1752年同步以月亮为参照的立法

练习深入思考：

1. 如何用等价类划分方法设计公历日期程序的测试用例？
2. 如何结合等价类划分和边界值分析设计公历日期程序的测试用例？

黑盒测试方法——边界值分析

■ 边界值分析原理

次边界条件/隐性边界：产品说明书中没有，外部用户看不到

1. 2的幂：各种数据类型的极限值
2. ASCII表，Unicode字符总数：
3. 空白 空值 零值 无输入等条件



廖力

黑盒测试方法——边界值分析

■ 边界值分析总结

1. 边界值需要彻底测试
2. 考虑资源极限，如有限的缓存
3. 需求规格中对硬件资源的限制
4. 输出值的边界也需要考虑



廖力

黑盒测试方法——边界值分析

■ 边界值分析和等价类划分的关系

等价类划分时，往往先要确定边界值。
边界值分析是等价类划分方法的补充。
测试中需要将两者结合起来使用。



廖 力

黑 盒 测 试 方 法 — 因 果 分 析 法



廖 力

东南大学 软件学院 College of Software Engineering

2019年10月21日

黑盒测试方法——因果分析法

■ 因果分析法背景

测试挑战：多个输入条件的关联问题

例1：火车票价查询：终点、车次、硬座/卧铺、普通票/学生票等多个因素决定

例2：包裹费用计算：快递/EMS、重量、距离等

组合测试面临处理大量测试用例的现实



廖力

黑盒测试方法——因果分析法

■ 因果分析法基础

因果图（Cause and Effect Diagram）：

将导致问题的原因划分为多种因素，并描述这些因素间的关系，从而找出问题根源的复杂问题分析工具。

因果图在管理学中被广泛使用，也被称为鱼骨图
(Fishbone Diagram)或石川图(Ishikawa Diagram)

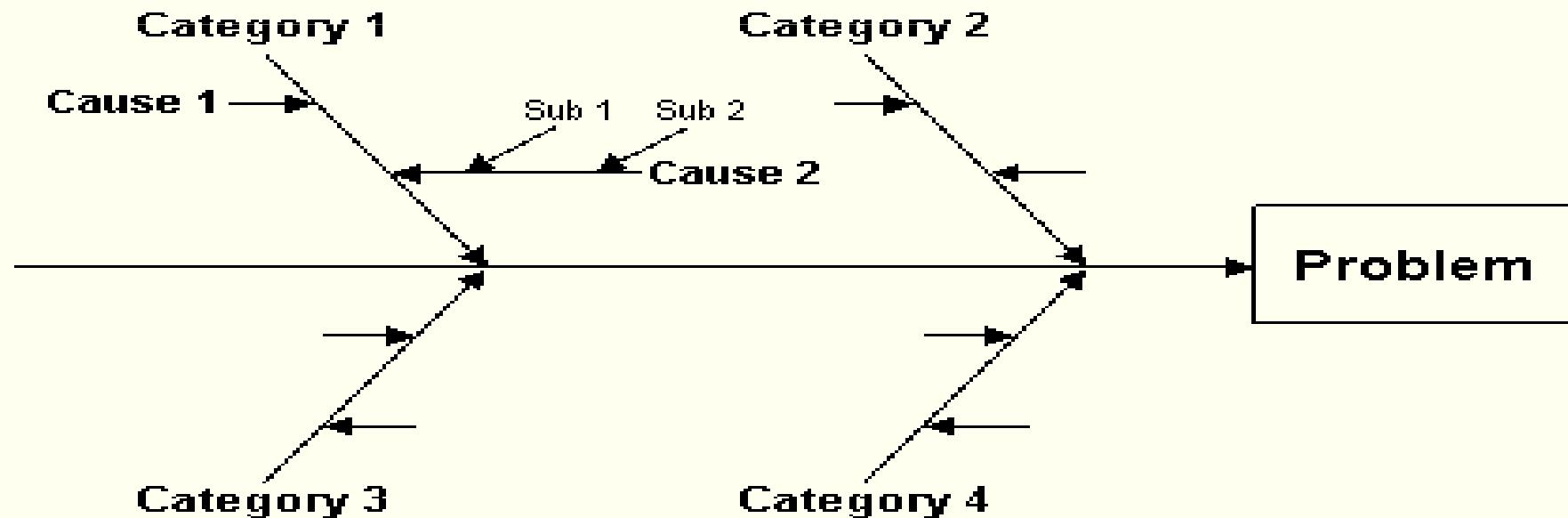


廖力

黑盒测试方法——因果分析法

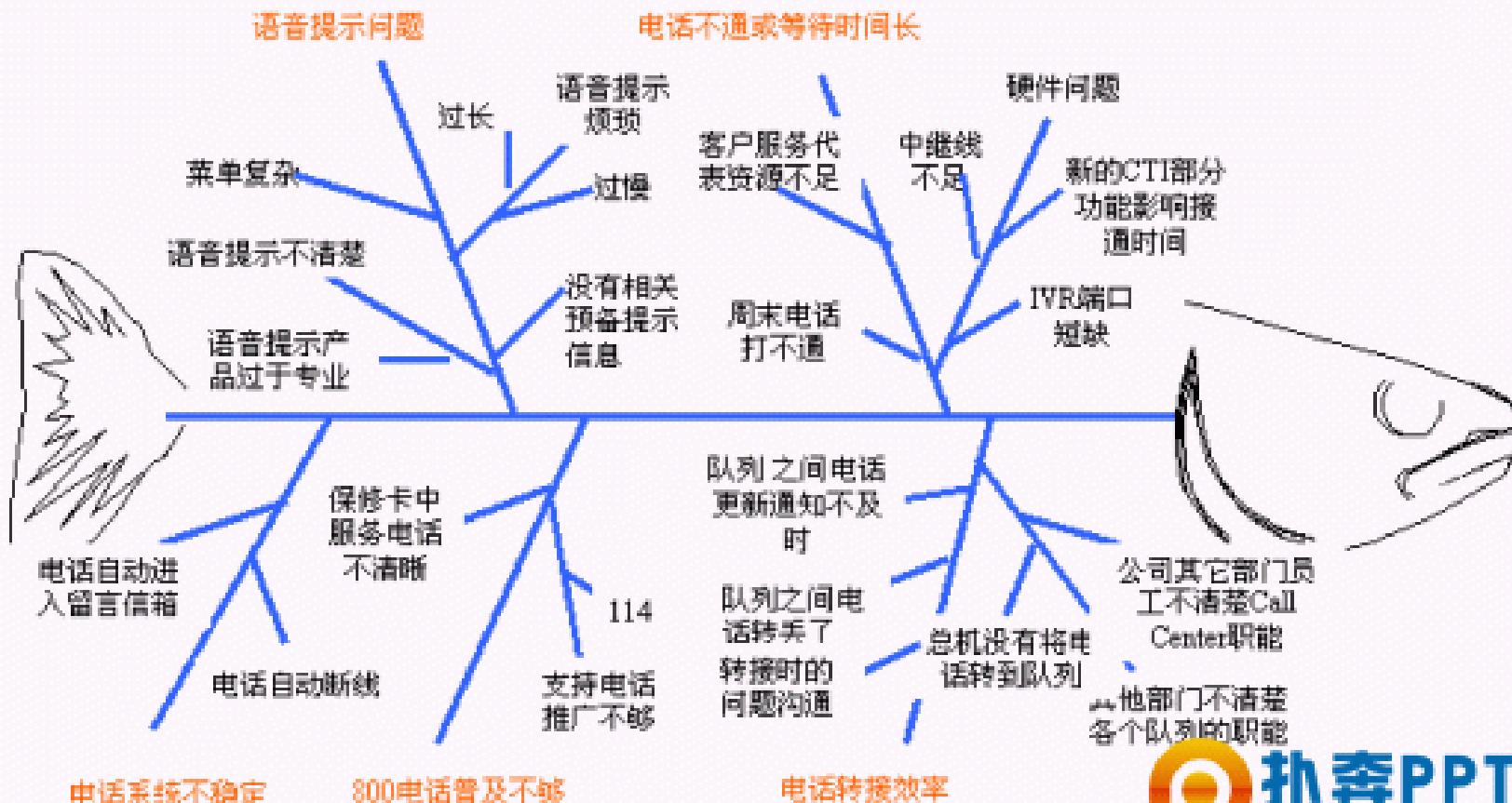
■ 因果分析法基础

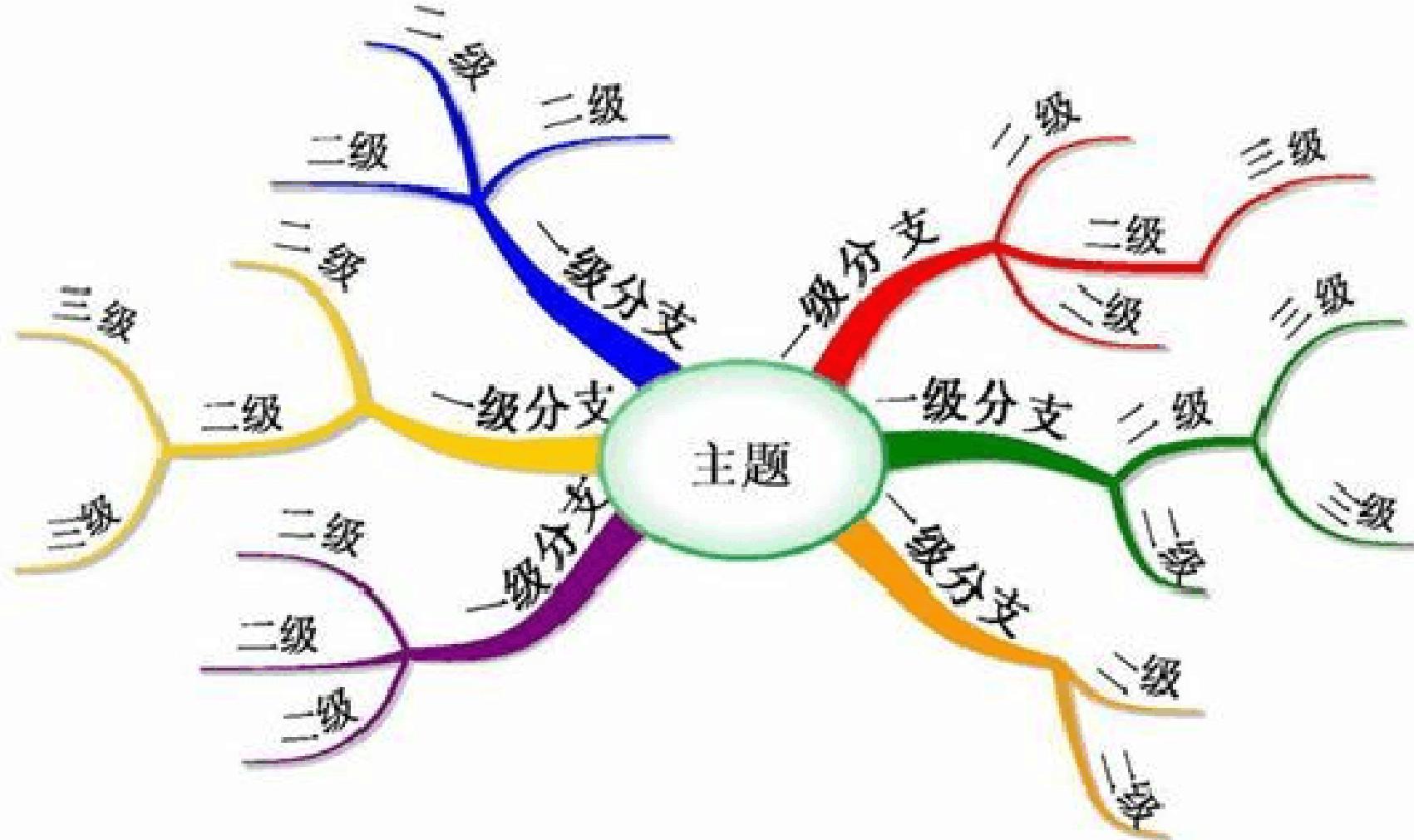
Ishikawa (Fishbone) Diagram



CallCenter 系统分析

鱼骨图分析





廖力

软件

Software Testing



UNIVERSITY

廖力

2019年10月21日

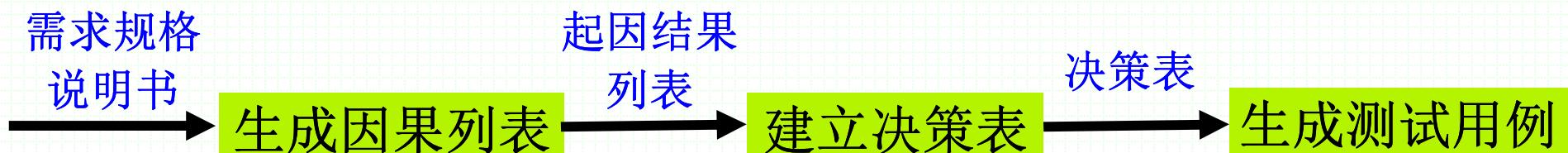
黑盒测试方法——因果分析法

■ 因果分析法基础

原理：

- (1) 软件的输入和输出之间存在逻辑关系，即因果图；
- (2) 因果图可从规格说明书中获得

过程：



廖力

黑盒测试方法——因果分析法

■ 因果图的表示

符号： C_i 表示原因 E_i 表示结果

原因和结果的4种关系：

(1) 恒等 (—) :

若 C_i 出现，则 E_i 出现；若 C_i 不出现，则 E_i 也不出现

恒等 (—)



廖力

黑盒测试方法——因果分析法

■ 因果图的表示

符号： C_i 表示原因 E_i 表示结果

原因和结果的4种关系：

(2) 非 (\sim) :

若C出现，则E不出现；若C不出现，则E出现

非 (\sim)



廖 力

黑盒测试方法——因果分析法

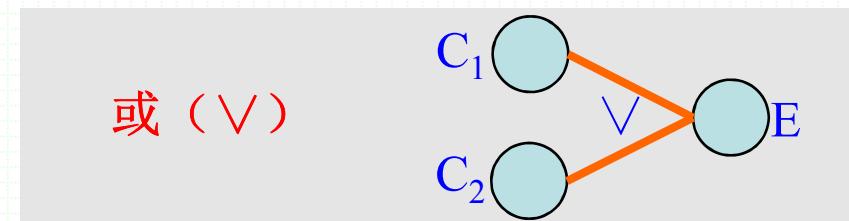
■ 因果图的表示

符号： C_i 表示原因 E_i 表示结果

原因和结果的4种关系：

(3) 或 (\vee) :

若几个 C_i 中有一个出现，则 E 出现；若几个 C_i 都不出现，则 E 不出现



廖力

黑盒测试方法——因果分析法

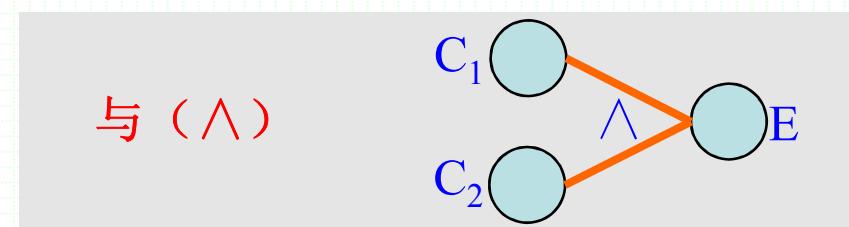
■ 因果图的表示

符号： C_i 表示原因 E_i 表示结果

原因和结果的4种关系：

(4) 与 (\wedge) :

若几个 C_i 都出现，则结果出现；若其中一个 C_i 不出现，则 E 不出现



廖力

黑盒测试方法——因果分析法

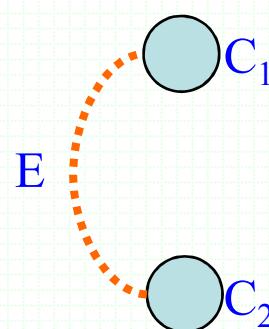
■ 因果图的表示

4种输入约束：

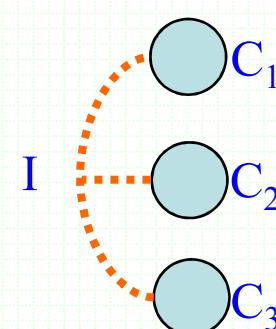
(1) 互斥(E): 多个原因不能同时成立，最多有一个能成立；
即 C_i 不能同时为1

(2) 包含(I): 多个原因中至少有一个必须成立；即 C_i 不能同时为0

互斥(E)



包含(I)



廖力

黑盒测试方法——因果分析法

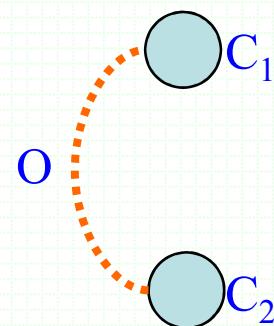
■ 因果图的表示

4种输入约束：

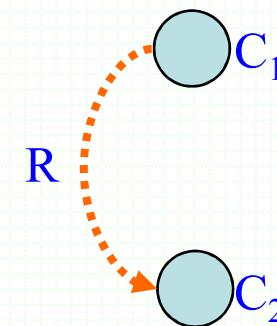
(3) 唯一(O)：多个原因中必须有一个且只有一个成立；即
 C_i 只有一个为1

(4) 要求(R)：当 C_1 成立， C_2 也必须成立；

唯一(O)



要求(R)



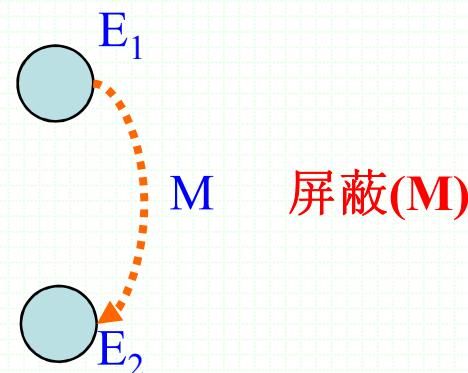
廖力

黑盒测试方法——因果分析法

■ 因果图的表示

1种输出约束：

- (1) 屏蔽(M)：当 E_1 是1时， E_2 必须是0；当 E_1 是0， E_2 的值不定；



廖力

黑盒测试方法——因果分析法

■ 因果图分析练习1

文件管理系统规格说明

- (1) 文件第一列的字符必须是一个A或B，且文件第二列的字符必须是一个数字；
- (2) 若符合上述情况，文件是被修改过；
- (3) 第一个字符不正确，则打印X12消息；
- (4) 第二个不是数字，则打印X13消息；

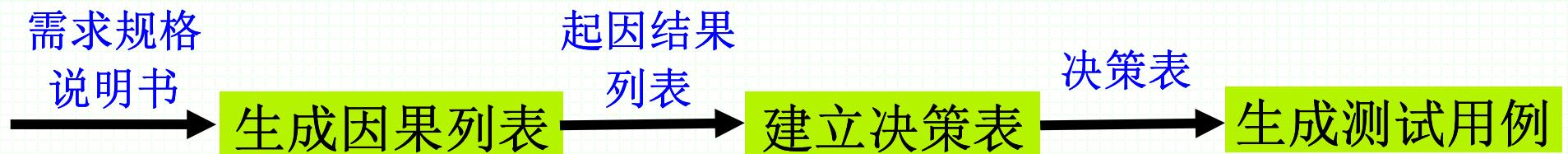


廖 力

黑盒测试方法——因果分析法

■ 因果分析法基础

过程：



廖 力

黑盒测试方法——因果分析法

■ 生成因果图

原因：

C1: 第1列的字符是A

C2: 第1列的字符是B

C3: 第2列的字符是数字

结果：

E1: 文件修改过

E2: 打印消息X12

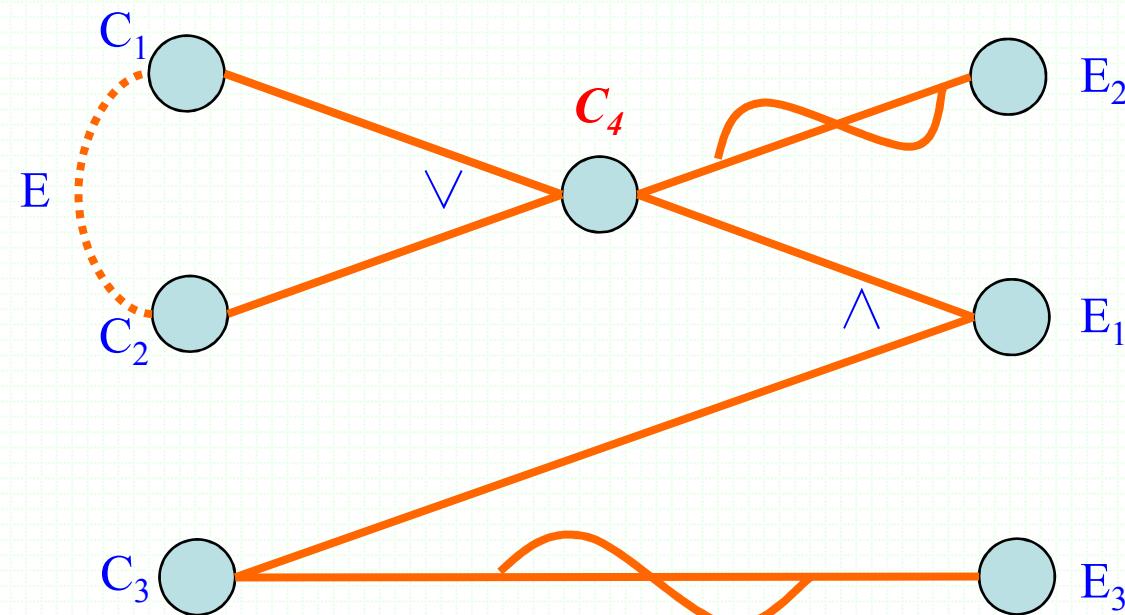
E3: 打印消息X13

词 力



黑盒测试方法——因果分析法

■ 生成因果图



廖 力

黑盒测试方法——因果分析法

■ 因果图列表

因果图列表：原因的所有组合及相应的结果组合

注意：某些原因的组合不存在

因果图列表示例

		1	2	3	4	5	6	7	8
输入（原因）	C1	0	0	0	0	1	1	1	1
	C2	0	0	1	1	0	0	1	1
	C3	0	1	0	1	0	1	0	1
输出（结果）	E1	0	0	0	1	0	1		
	E2	1	1	0	0	0	0		
	E3	1	0	1	0	1	0		



黑盒测试方法——因果分析法

■ 生成决策表

方法：

原因——决策表中的条件 结果——决策表中的行动

原因与结果的组合——决策规则

决策表示例

		1	2	3	4	5	6
条件	C1	0	0	0	0	1	1
	C2	0	0	1	1	0	0
	C3	0	1	0	1	0	1
行动	A1	0	0	0	1	0	1
	A2	1	1	0	0	0	0
	A3	1	0	1	0	1	0

廖 力



黑盒测试方法——因果分析法

■ 生成测试用例

- (1) 决策表中的条件——测试用例的输入条件
- (2) 决策表中的行动——测试用例的期望输出
- (3) 一条决策规则——一个测试用例



廖力

黑盒测试方法——因果分析法

■ 生成测试用例

		1	2	3	4	5	6
条件	C1	0	0	0	0	1	1
	C2	0	0	1	1	0	0
	C3	0	1	0	1	0	1
行动	A1	0	0	0	1	0	1
	A2	1	1	0	0	0	0
	A3	1	0	1	0	1	0
用例	输入	DY	C2	BN	B5	AM	A3
	输出						



廖 力

黑盒测试方法——因果分析法

■ 因果图分析练习2

饮料自动售货机规格说明

- (1) 若投入5角钱或1元钱的硬币，压下【橙汁】或【啤酒】的按钮，则相应的饮料就送出来。
- (2) 若售货机没有零钱找，则一个显示【零钱找完】的红灯亮，这时在投入1元硬币并压下按钮后，饮料不送出来而且1元硬币也退出来；
- (3) 若有零钱找，则【零钱找完】的红灯灭，在送出饮料的同时退还5角硬币。



廖力

黑盒测试方法——因果分析法

■ 生成因果图

原因：

C1:售货机有零钱找

C2:投入1元硬币

C3:投入5角硬币

C4:压下橙汁按钮

C5:压下啤酒按钮



廖 力

黑盒测试方法——因果分析法

■ 生成因果图

结果：

E1:售货机『零钱找完』灯亮

E2:退还1元硬币

E3:退还5角硬币

E4:送出橙汁饮料

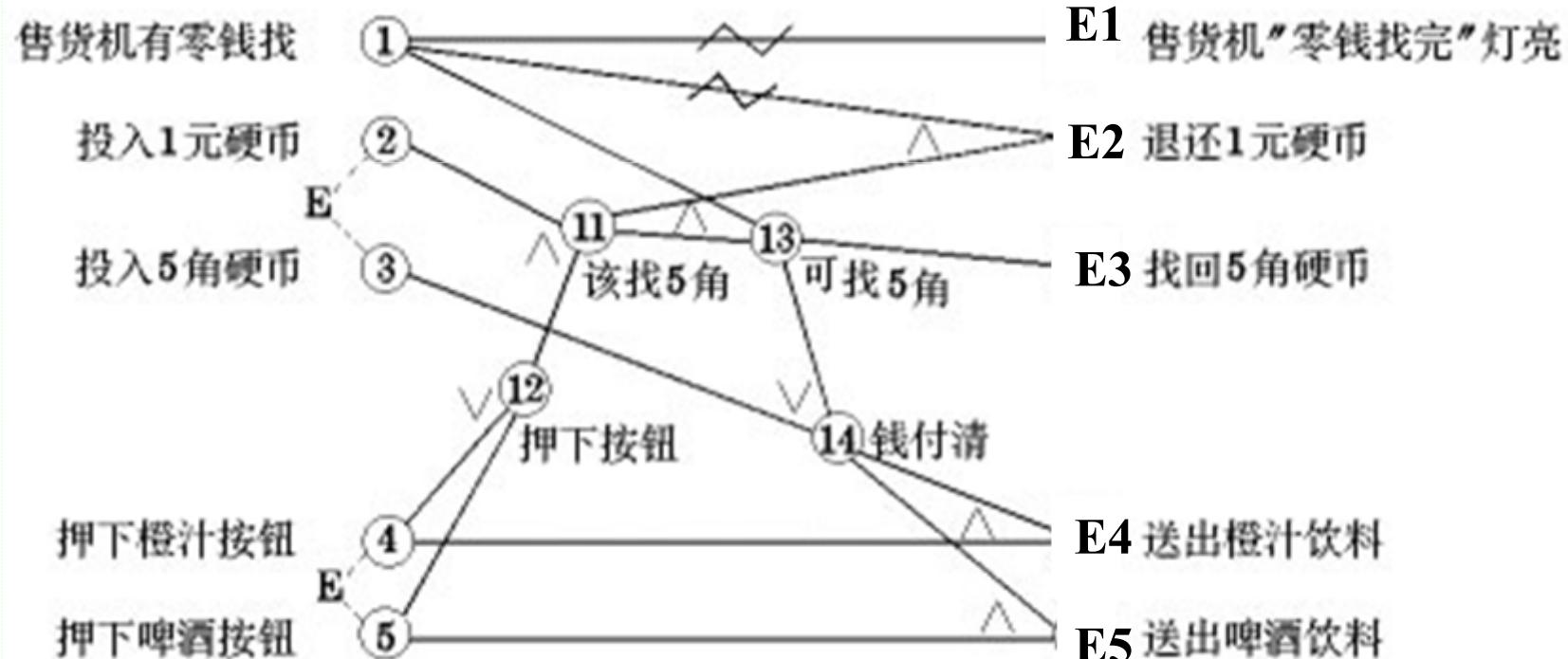
E5:送出啤酒饮料



廖 力

黑盒测试方法——因果分析法

■ 生成因果图



黑盒测试方法——因果分析法

■ 生成决策表和测试用例

序号	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1	2						
条 件	①	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
	②	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0					
	③	1	1	1	1	0	0	0	1	1	1	0	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	0	0	0	0	0	0					
	④	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0					
	⑤	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0					
中 间 结 果	⑪					1	1	0		0	0	0	0	0	0					1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
	⑫					1	1	0		1	1	0		1	1	0				1	1	0	1	1	0	1	1	0										
	⑬					1	1	0		0	0	0		0	0	0				0	0	0	0	0	0	0	0											
	⑭					1	1	0		1	1	1		0	0	0				0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0		
结 果	⑯					0	0	0		0	0	0		0	0	0				1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
	⑰					0	0	0		0	0	0		0	0	0				1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	⑱					1	1	0		0	0	0		0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	⑲					1	0	0		1	0	0		0	0	0				0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	⑳					0	1	0		0	1	0		0	0	0				0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
测试 用例						Y	Y	Y		Y	Y	Y		Y	Y	Y				Y	Y	Y	Y	Y	Y	Y	Y											

C1:售货机有零钱找

C2:投入1元硬币

C3:投入5角硬币

C4:按下橙汁按钮

C5:按下啤酒按钮

阴影部分表示因违反约束条件的不可能出现的情况，删去。

第16列与第32列因什么动作也没做，也删去。

最后可根据剩下的16列作为确定测试用例的依据



廖 力

2019年10月21日

黑盒测试方法——因果分析法

■ 因果分析法总结

Step1: 分析规格说明书，识别原因和结果

Step2: 在因果图连接原因和结果

Step3: 标明原因之间以及结果之间的约束条件

Step4: 因果图转换为因果图列表进而生成决策表

Step5: 决策表的规则转换为测试用例



廖力

黑 盒 测 试 方 法 — 决 策 表



廖 力

东南大学 软件学院 College of Software Engineering

2019年10月21日

黑盒测试方法——决策表

■ 决策表方法原理

决策表并非因果图的一个辅助工具

适用情形：

输入输出较多且输入之间和输出之间互相制约的
条件较多

决策表：

把作为条件的**所有输入的组合**以及**对应输出都罗**
列出来形成的表格。

特点：

能将复杂问题按照各种可能情况全部列出，表示
简明，并避免遗漏



廖 力

黑盒测试方法——决策表

■ 生成决策表

序号	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1	2					
条件	①	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
	②	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0					
	③	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0	1	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0					
	④	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0					
	⑤	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1					
中间结果	⑪					1	1	0		0	0	0	0	0	0					1	1	0	0	0	0	0	0	0	0	0	0	0	0	0			
	⑫						1	1	0		1	1	0		1	1	0			1	1	0	1	1	0		1	1	0								
	⑬							1	1	0		0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
	⑭								1	1	0		1	1	1		0	0	0		0	0	0	1	1	1		0	0	0	0	0	0	0	0		
结果	⑯									0	0	0		0	0	0					1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
	⑰									0	0	0		0	0	0					1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	⑱									1	1	0		0	0	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	⑲									1	0	0		1	0	0		0	0		0	0	0	1	0	0	0	0	0	0	0	0	0	0	0		
	⑳									0	1	0		0	1	0		0	0		0	0	0	0	1	0	0	0	0	0	0	0	0	0	0		
测试用例										Y	Y	Y		Y	Y	Y					Y	Y	Y	Y	Y	Y											

- C1:售货机有零钱找
- C2:投入1元硬币
- C3:投入5角硬币
- C4:压下橙汁按钮
- C5:压下啤酒按钮



廖 力

2019年10月21日

黑盒测试方法——决策表

■ 决策表组成

4种成分：

- (1) 条件桩：列出所有可能问题（条件）
- (2) 条件项：列出条件所有可能取值
- (3) 动作桩：列出可能采取的操作
- (4) 动作项：指出在条件项的各种取值情况下应采取的动作

决策规则：贯穿条件项和动作项的一列



廖力

决策规则

序号	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	30	1	2	
条 件 桩 条件项	①	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
中 间 结 果	②	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	0	0
结 果 动作项	③	1	1	1	1	0	0	0	0	1	1	1	0	0	0	1	1	1	1	1	0	0	0
结 果 动作桩	④	1	1	0	0	1	1	0	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1
结 果 用例	⑤	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
测 试 用例	⑪										1	1	0					1	1	0	0	0	
	⑫										1	1	0					1	1	0	1	1	
	⑬										1	1	0					0	0	0	0	0	
	⑭										1	1	0					0	0	0	1	1	
结 果 用例	⑯										0	0	0					1	1	1	1	1	
结 果 用例	⑰										0	0	0					1	1	0	0	0	
结 果 用例	⑱										1	1	0					0	0	0	0	0	
结 果 用例	⑲										1	0	0					0	0	0	1	0	
结 果 用例	⑳										0	1	0					0	0	0	0	0	



黑盒测试方法——决策表

■ 决策表构造

Step1: 列出所有的条件桩和动作桩

Step3: 填入条件项

Step4: 填入动作项，得到初始决策表

Step5: 简化决策表，合并相似规则

规则可能总数： 2^n



廖 力

黑盒测试方法——决策表

■ 例如：

对功率大于**100**马力的机器、维修记录不全或已运行**10**年以上的机器，应给予优先的维修处理。

条件1：机器功率大小

条件2：维修记录

条件3：运行时间

动作1：机器能否得到优先维修



廖 力

黑盒测试方法——决策表

■ 决策表构造

初始化决策表

		1	2	3	4	5	6	7	8
条件	功率大于100马力	Y	Y	Y	Y	N	N	N	N
	维修记录不全	Y	Y	N	N	Y	Y	N	N
	运行时间超过10年	Y	N	Y	N	Y	N	Y	N
动作	优先维修	√	√	√	√	√	√	√	
	正常维修								√



廖力

黑盒测试方法——决策表

■ 决策表化简

简化目标：合并相似规则

相似规则判断：

有两条或以上规则具有相同动作，并且在条件项之间存在极大相似，便可以合并

“—”：

表示合并后该条件项与取值无关，称“无关条件”



廖力

黑盒测试方法——决策表

■ 决策表构造

		初始化决策表							
		1	2	3	4	5	6	7	8
条件	功率大于100马力	Y	Y	Y	Y	N	N	N	N
	维修记录不全	Y	Y	N	N	Y	Y	N	N
	运行时间超过10年	Y	N	Y	N	Y	N	Y	N
动作	优先维修	✓	✓	✓	✓	✓	✓	✓	
	正常维修								✓



廖力

黑盒测试方法——决策表

■ 决策表化简

初始化决策表化简

		1	2	3	4	5	6	7	8
条件	功率大于100 马力	Y		N		N		N	
	维修记录不全	—		Y		N		N	
	运行时间超过 10年	—		—		Y		N	
动作	优先维修	✓		✓		✓			
	正常维修							✓	



廖力

黑盒测试方法——因果分析法

■ 生成决策表和测试用例

序号	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1	2
条件	①	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	②	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
	③	1	1	1	1	0	0	0	1	1	1	0	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	0	0	0	0	0
	④	1	1	0	0	1	1	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	
	⑤	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	
中间结果	⑪					1	1	0		0	0	0	0	0						1	1	0	0	0	0	0	0	0	0	0	0	
	⑫					1	1	0		1	1	0	1	1	0					1	1	0	1	1	0	1	1	0	0	0	0	
	⑬					1	1	0		0	0	0	0	0						0	0	0	0	0	0	0	0	0	0	0	0	
	⑭					1	1	0		1	1	1	0	0						0	0	0	1	1	1	0	0	0	0	0	0	
结果	⑯					0	0	0		0	0	0	0	0						1	1	1	1	1	1	1	1	1	1	1	1	
	⑰					0	0	0		0	0	0	0	0						1	1	0	0	0	0	0	0	0	0	0	0	
	⑱					1	1	0		0	0	0	0	0						0	0	0	0	0	0	0	0	0	0	0	0	
	⑲					1	0	0		1	0	0	0	0						0	0	0	1	0	0	0	0	0	0	0	0	
	⑳					0	1	0		0	1	0	0	0						0	0	0	0	1	0	0	0	0	0	0	0	
测试用例						Y	Y	Y		Y	Y	Y							Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		



C1:售货机有零钱找

C2:投入1元硬币

C3:投入5角硬币

C4:压下橙汁按钮

C5:压下啤酒按钮
E21:售货机【零钱找完】灯亮

E22:退还1元硬币

E23:退还5角硬币

E24:送出橙汁饮料

E25:送出啤酒饮料

黑盒测试方法

- 1. 等价类划分
- 2. 边界值分析
- 3. 因果分析法
- 4. 决策表
- 5. 基于图/状态的测试
- 6. 正交数组测试



廖 力

黑盒测试方法 —基于模型的测试



廖力

东南大学 软件学院 College of Software Engineering

2019年10月21日

黑盒测试方法——基于状态/图的测试

■ 基于模型的测试原理

模型：

1. 理解复杂系统的工作过程
2. 解释没有完全理解的事物
3. 将抽象的事物变具体

建模过程迫使我们将复杂的事物分解成许多清晰而且易理解的单元，最终让我们更好地理解整个问题



廖 力

黑盒测试方法——基于状态/图的测试

■ 基于模型的测试简介

原理：

1. 软件执行过程可分解为若干对象和连接对象间的关系
2. 测试序列可视为验证对象间所期望的关系是否满足



廖力

黑盒测试方法——基于状态/图的测试

■ 基于模型的测试简介

适用领域：

1. **有限状态建模**：软件产品类似语言处理器，拥有类似语言语法的状态机或上下文无关文法
2. **工作流建模**：根据当前状态和输入，执行具体工作流，产生新输出和新状态
3. **数据流建模**：系统建模为一组数据流，并从一个状态转换到另一个状态
4. **时间建模**：对象间具有顺序连接，连接权值表示执行时间



廖 力

黑盒测试方法——基于模型的测试

■ 模型化软件

1. 图论

例如：欧拉的“哥尼斯堡七桥问题”

2. 工作流

例如：业务流程

3. 语法模型

例如：手机号码

4. 其它：贝叶斯模型、Petri网、……



廖 力

黑盒测试方法——基于状态/图的测试

■ 模型化软件—图论



表示GUI的状态图



廖 力

东南大学 软件学院 College of Software Engineering

2019年10月21日

黑盒测试方法——基于状态/图的测试

■ 模型化软件—图论

测试用例

0. 启动“计算器”
1. 输入 $1+1$
2. 显示结果2
3. 清除结果
4. 计算器归零
5. 停止应用程序

0. 启动“计算器”
1. 选择“查看”--“科学型”
2. 科学型界面
3. 选择“查看”--“标准型”
4. 标准型界面
5. 停止应用程序

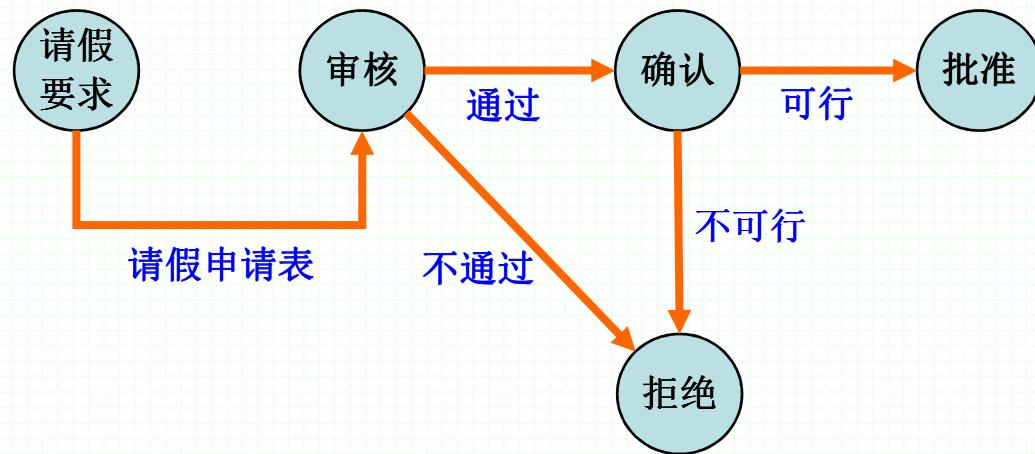


廖力

黑盒测试方法——基于状态/图的测试

■ 模型化软件——工作流

节点：对象 边：关系



表示工作流的状态图



廖力

黑盒测试方法——基于模型的测试

■ 基于模型方法练习

输入数字具有这样的特点：可带+/-符号，可以是小数，但小数位数最多为2位，以#号为结束标记

请用基于模型的方法设计该输入的测试用例

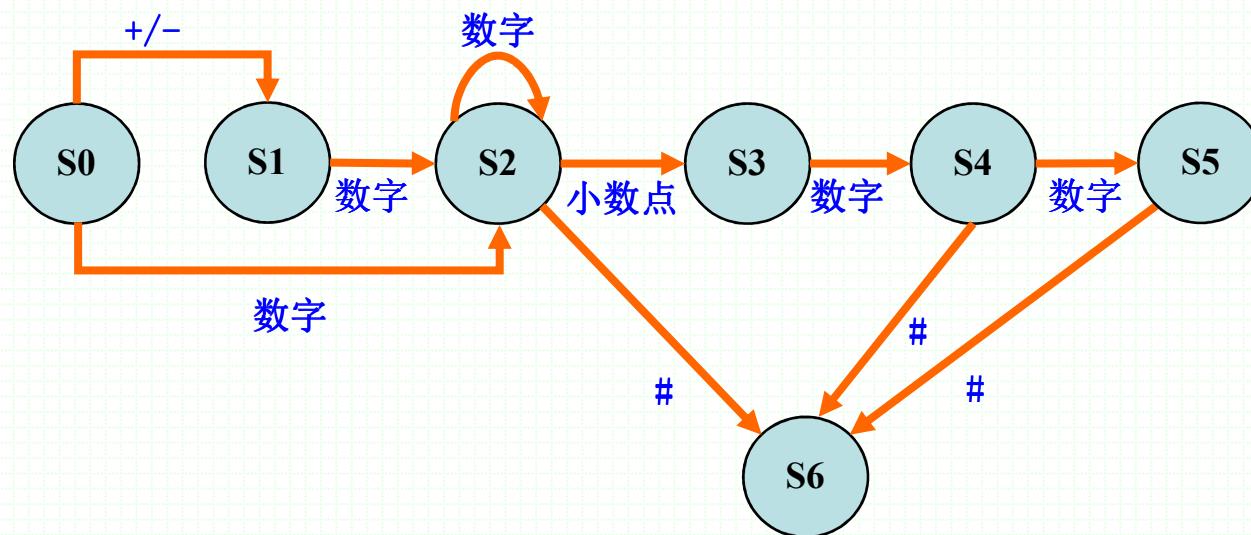


廖 力

黑盒测试方法——基于状态/图的测试

■ 模型化软件

节点：对象 边：关系



表示状态机的状态图

确认数字有效性

1. 数字可以带符号
2. 符号后可接任意位数字
3. 可以有小数点
4. 小数点位数最多为2位
5. 任何数字都以#结束



廖 力

黑盒测试方法——基于状态/图的测试

■ 基于模型的测试简介

测试用例生成：

遍历图，并覆盖
图中所有的边，
就可以导出测
试用例

测试用例生成举例

当前状态	输入	期望状态
0	+	1
0	2	2
1	2	2
2	3	2
2	.	3
3	1	4
4	4	5
5	#	6
...	



廖 力

黑 盒 测 试 方 法 —正交数组测试



廖 力

东南大学 软件学院 College of Software Engineering

2019年10月21日

在测试中,特别是互联网应用,我们无法规定用户的环境,分布在世界各地的用户,其使用的环境是各种各样的.例如:

- 操作系统: **Windows Vista, Windows XP Home, Windows XP professional, Windows 2000 pro, Windows 2000 server, Windows NT, Windows 98, Linux, Solaris 9, Solaris 10, Mac OS 9, Mac OS X**
- 浏览器: **IE 6.0, IE 7.0, FireFox 1.5, FireFox 2.0, 遨游、腾讯、苹果etc。**
- 代理服务器/防火墙: **ISA 2000, ISA 2004, Blue Coast, Cisco PIX, Linux squid, Checkpoint, ...**
- 防火墙验证方式: 无口令, 口令, **Script, ...**
- 传输协议: **TCP, HTTP, SSL**



- 假如：
 - 操作系统 – 12
 - 浏览器 - 9
 - 代理服务器/防火墙 - 6
 - 防火墙验证方式 - 3
 - 传输协议 – 3
 - 客户端版本 – 3
 - 第3方产品集成, **Lotus Notes** – 6
 - 本地化语言 – 选 4 种：英文(**EL**)、中文繁体(**B5**)、日文 (**JP**) 、德文(**GE**)



廖 力

- 如果用一个完全的组合，将是爆炸性的组合，测试工作量将非常大。其完全组合数是：

$$12 \times 9 \times 6 \times 3 \times 3 \times 3 \times 6 \times 4 = \\ 419904$$

- 假如每个组合产品的功能测试用例为1000个，则需执行 $419904 \times 1000 \rightarrow$ 4亿多测试用例



廖 力

正交试验设计方法

- 正交试验设计方法是从大量的试验数据中挑选适量的、有代表性的点，从而合理地安排测试的一种科学的试验设计方法
- 使用已经造好了的表格——正交表来安排试验并进行数据分析。



廖 力

- 1992年，AT&T对PC(IBM 格式)和StarMail(基于局域网的电子邮件软件)做回归测试；
- 最初制定的测试计划是用18周的时间执行1500个测试用例。
- 计划2，测试时间8周，2个人8周的时间测试1000个测试用例，但无法保证测试的质量。
- 为了减轻这种不确定性的问题，用正交表法重新设计了测试用例，此时测试用例只有422个。发现了41个缺陷。
- 在使用的两年时间内，凡被测试到的领域都没有再发现缺陷。



廖力

- 据测试负责人估计，如果AT&T采用1000个测试用例的测试计划，可能仅仅只发现这些缺陷中的32个。
- 与计划2相比，用正交表设计测试用例执行工作量不到50%，但却多发现28%的缺陷
-



黑盒测试方法——正交数组测试

■ 正交数组测试

案例： 114系统查询测试

当用户打114 查询某公司的电话时，电信局的坐席人员会输入该公司相关信息，并进行查询，最后把查询的结果告之用户。界面如下：

单位基本信息查询		查询参数：音形码[;类别码附属名] 拼音码[;类别码附属名] 范名码 行业类别 特征码	
音形码[;类别码附属名][F7]	<input type="text"/>	拼音码[;类别码附属名][F11]	<input type="text"/>
范名码[F9]	<input type="text"/>	行业类别[F12]	<input type="text"/>
特征码[F8]	<input type="text"/>		



廖 力

黑盒测试方法——正交数组测试

■ 正交数组测试

案例： 114系统查询测试

测试方法

- 全部测试
- 部分测试
- 用正交表法设计用例并测试



廖 力

软件测试

Software Testing

案例： 114系统查询测试

测试方法

- 全部测试： 32个测试用例

序号	音形码	拼音码	路名码	行业类别	特征码
1	不填	不填	不填	不填	不填
2	不填	不填	不填	不填	填
3	不填	不填	不填	填	不填
4	不填	不填	不填	填	填
5	不填	不填	填	不填	不填
6	不填	不填	填	不填	填
7	不填	不填	填	填	不填
8	不填	不填	填	填	填
9	不填	填	不填	不填	不填
10	不填	填	不填	不填	填
11	不填	填	不填	填	不填
12	不填	填	不填	填	填
13	不填	填	填	不填	不填
14	不填	填	填	不填	填
15	不填	填	填	填	不填
16	不填	填	填	填	填
17	填	不填	不填	不填	不填
18	填	不填	不填	不填	填
19	填	不填	不填	填	不填
20	填	不填	不填	填	填
21	填	不填	填	不填	不填
22	填	不填	填	不填	填
23	填	不填	填	填	不填
24	填	不填	填	填	填
25	填	填	不填	不填	不填
26	填	填	不填	不填	填
27	填	填	不填	填	不填
28	填	填	不填	填	填
29	填	填	填	不填	不填
30	填	填	填	不填	填
31	填	填	填	填	不填
32	填	填	填	填	填

软件测试

Software Testing

案例： 114系统查询测试

测试方法

- 部分测试：面临如何选择以及测不全的问题。

序号	音形码	拼音码	路名码	行业类别	特征码
1	0	0	0	0	0
2	0	0	0	0	1
3	0	0	0	1	0
4	0	0	0	1	1
5	0	0	1	0	0
6	0	0	1	0	1
7	0	0	1	1	0
8	0	0	1	1	1
9	0	1	0	0	0
10	0	1	0	0	1
11	0	1	0	1	0
12	0	1	0	1	1
13	0	1	1	0	0
14	0	1	1	0	1
15	0	1	1	1	0
16	0	1	1	1	1
17	1	0	0	0	0
18	1	0	0	0	1
19	1	0	0	1	0
20	1	0	0	1	1
21	1	0	1	0	0
22	1	0	1	0	1
23	1	0	1	1	0
24	1	0	1	1	1
25	1	1	0	0	0
26	1	1	0	0	1
27	1	1	0	1	0
28	1	1	0	1	1
29	1	1	1	0	0
30	1	1	1	0	1
31	1	1	1	1	0
32	1	1	1	1	1

软件测试

Software Testing

案例： 114系统查询测试

测试方法

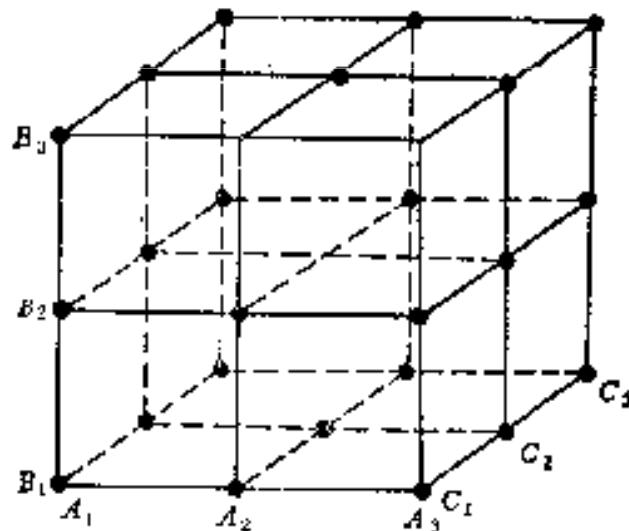
•正交测试：利用正交测试再加上特殊用例测试，基本上可以均匀分散地覆盖到各种情况。又能大大减少测试用例个数。

序号	音形码	拼音码	路名码	行业类别	特征码	ITY
1	0	0	0	0	0	
2	0	0	0	0	1	
3	0	0	0	1	0	
4	0	0	0	1	1	
5	0	1	0	0	0	
6	0	1	1	0	1	
7	0	1	1	1	0	
8	0	0	1	1	1	
9	0	1	0	0	0	
10	0	1	0	0	1	
11	0	1	0	1	0	
12	0	1	0	1	1	
13	0	1	1	0	0	
14	0	1	1	0	1	
15	0	1	1	1	0	
16	0	1	1	1	1	
17	1	0	0	0	0	
18	1	0	0	0	1	
19	1	0	0	1	0	
20	1	0	0	1	1	
21	1	0	1	0	0	
22	1	0	1	0	1	
23	1	0	1	1	0	
24	1	0	1	1	1	
25	1	1	0	0	0	
26	1	1	0	0	1	
27	1	1	0	1	0	
28	1	1	0	1	1	
29	1	1	1	0	0	
30	1	1	1	0	1	
31	1	1	1	1	0	
32	1	1	1	1	1	

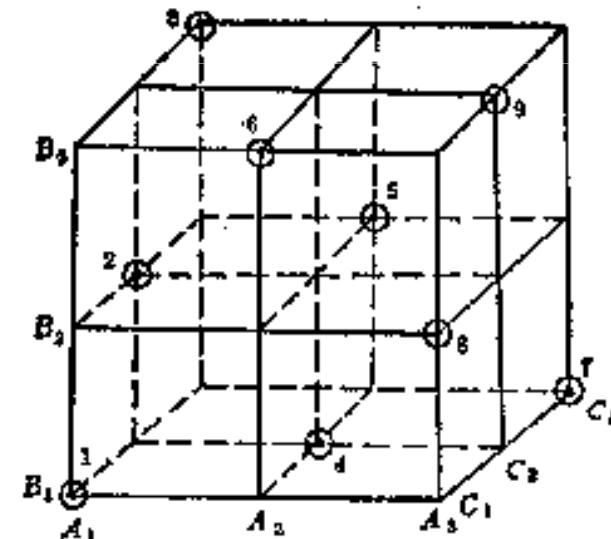
黑盒测试方法——正交数组测试

■ 正交数组测试原理

输入：A, B, C，每个输入有三种取值可能



全面测试: $3 \times 3 \times 3 = 27$



正交测试: $3 \times 3 = 9$



廖 力

黑盒测试方法——正交数组测试

■ 正交数组测试原理

正交测试用例：

A₁ B₁ C₁

A₁ B₂ C₂

A₁ B₃ C₃

A₂ B₁ C₂

A₂ B₃ C₁

A₂ B₂ C₃

A₃ B₁ C₃

A₃ B₂ C₁

A₃ B₃ C₂

任意两个输入的所有组合均被包含



廖 力

黑盒测试方法——正交数组测试

■ 正交数组测试

正交试验设计是研究多因素多水平的一种设计方法，它是根据正交性从全面试验中挑选出部分有代表性的点进行试验，这些有代表性的点具备了“均匀分散，齐整可比”的特点



廖力

黑盒测试方法——正交数组测试

■ 正交数组测试背景

拉丁方：

古希腊，国王在检阅臣民方队时，要求每行有一个民族的代表，每列也要有一个民族的代表。

1	2	3
2	3	1
3	1	2

n阶拉丁方：用n个不同的拉丁字母排成一个n阶方阵（n<26），如果每行的n个字母均不相同，每列的n个字母均不相同，则称这种方阵为n*n拉丁方或n阶拉丁方。每个字母在任一行、任一列中只出现一次。



廖 力

黑盒测试方法——正交数组测试

■ 正交数组测试背景

Galois理论：

在科学实验中，可以从大量实验数据中挑选适量的，有代表性的点，从而合理地安排实验的一种科学实验方法。

http://en.wikipedia.org/wiki/Galois_theory



Évariste Galois (1811-1832)



廖力

黑盒测试方法——正交数组测试

■ 正交数组测试原理

正交数组测试：

从大量测试用例中挑选适量的，有代表性的用例，从而合理地进行测试的方法。

适用领域：

输入域相对较少而详尽测试的次数又过大的问题

特点：

输入域有限，参数取值有限，需要将参数取值进行排列组合



廖 力

黑盒测试方法——正交数组测试

■ 正交表

一、正交表的构成

因子(Factor):欲考察的变量因素（输入参数）

水平(Levels) : 单个因子的取值（输入取值）

因子数(Factors):正交表中列的个数

水平数(Levels):单个因子的取值个数

行数(Runs):正交表行数，即测试用例个数



廖 力

黑盒测试方法——正交数组测试

■ 正交表

二、正交表的形式

记法：L_{行数}(水平数因子数)

The diagram shows an orthogonal array L8(2⁷). It consists of 8 rows (行号) and 7 columns (列号). The columns are labeled 1 through 7. The first column is labeled "因子" (Factor). Red arrows point from the labels "因子" and "水平" to the first column and the second row respectively. The array contains binary values (0 or 1) representing different levels of factors.

行号	因子						
	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1
2	1	1	1	0	0	0	0
3	1	0	0	1	1	0	0
4	1	0	0	0	0	1	1
5	0	1	0	1	0	1	0
6	0	1	0	0	1	0	1
7	0	0	1	1	0	0	1
8	0	0	1	0	1	1	0

正交表可用查表或软件辅助来构造



廖力

黑盒测试方法——正交数组测试

■ 正交表

三、正交表的正交性

整齐可比性：

在同一张正交表中，**每个因子的每个水平出现的次数是完全相同的**。每个因子的每个水平与其它因子的每个水平参与试验的机率是完全相同的，这就保证在各个水平中最大程度排除了其它因子水平的干扰。

均衡分散性：

在同一张正交表中，**任意两列（两个因子）的水平搭配（横向形成的数字对）是完全相同的**。这样就保证了试验条件均衡地分散在因子水平的完全组合之中，因而具有很强的代表性，容易得到好的试验条件。

Experiment Number	Column 1	Column 2	Column 3
1	1	1	1
2	1	2	2
3	2	1	2
4	2	2	1



黑盒测试方法——正交数组测试

■ 正交表

四、如何得到正交表

正交表可用查表或软件
辅助来构造

<http://www.york.ac.uk/depts/mathstables/orthogonal.htm>

http://support.sas.com/techsup/technote/ts723_Designs.txt

		Taguchi Designs															
n\k	3	4	5	6	7	10	11	12	13	15	21	22	26	27	31	40	63
4		L4 2															
8						L8 2											
9			L9 3														
12							L12 2										
16				L16b 4									L16 2				
18						L18 3.6											
25					L25 5												
27								L27 3.2									
32						L32b 4.2							L32 2				
36											L36 3		L36b 3.2				
50								L50 5.2									
54													L54 3.2				
64										L64b 2				L64 4			
81													L81 3				



Orthogonal Arrays (Taguchi Designs)

L4: Three two-level factors

L8: Seven two-level factors

L9: Four three-level factors

L12: Eleven two-level factors

L16: Fifteen two-level factors

L16b: Five four-level factors

L18: One two-level and seven three-level factors

L25: Six five-level factors

L27: Thirteen three-level factors

L32: Thirty-two two-level factors

L32b: One two-level factor and nine four-level factors

.....



廖 力

软件测试

Software Testing

SOUTHEAST UNIVERSITY

Taguchi Designs

n\k	3	4	5	6	7	10	11	12	13	15	21	22	26	27	31	40	63
4	L4 2																
8										L8 2							
9		L9 3															
12										L12 2							
16			L16b 4									L16 2					
18												L18 3.6					
25						L25 5											
27												L27 3.2					
32							L32b 4.2							L32 2			
36										L36 3		L36b 3.2					
50											L50 5.2						
54												L54 3.2					
64												L64b 2				L64 4	
81														L81 3			

Experiment Number	Column		
	1	2	3
1	1	1	1
2	1	2	2
3	2	1	2
4	2	2	1

Experiment Number	Column						
	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1
2	1	1	1	2	2	2	2
3	1	2	2	1	1	2	2
4	1	2	2	2	2	1	1
5	2	1	2	1	2	1	2
6	2	1	2	2	1	2	1
7	2	2	1	1	2	2	1
8	2	2	1	2	1	1	2



廖 力

东南大学 软件学院 College of Software Engineering

2019年10月21日

黑盒测试方法——正交数组测试

■ 正交数组测试步骤

Step1: 确定因子和水平

有哪些因子？总因子数是多少？

每个因子有哪些取值？其水平数是多少？

Step2: 判断是否能使用正交数组

因子的个数少，如只有2个，则不适用

因子的个数较多时可考虑正交数组

Step3: 选择合适的正交表

根据因子数和水平数，选择测试次数最少的正交表



廖 力

黑盒测试方法——正交数组测试

■ 正交数组测试步骤

Step4: 把变量值映射到表中
填入输入变量的取值

Step5: 正交测试用例制作
每一行对应一个测试用例

Step6: 补充测试用例

根据需求规则说明书，增加一些正交表没有覆盖但需要测试的用例



廖 力

黑盒测试方法——正交数组测试

■ 选择正交表

选择原则：

- 考虑因子的个数
 - 正交表因子数 \geq 实际因子数
- 考虑因子的水平个数
 - 正交表每个因子数 \geq 实际每个因子数
- 考虑正交表的行数
 - 如果出现2个或2个以上正交表符合以上条件，则选择行数最少的正交表。



廖 力

黑盒测试方法——正交数组测试

■ 选择正交表

■ 没有和对应因子数对应的表？

包含的方法：取因子数最接近但略大的实际值的表

例：5因子2水平

满足的正交表有 $L_8(2^7)$ 和 $L_{16}(2^{15})$ ，取行数少的前者



廖力

软件

Software Testi

Taguchi Designs

NIVERSITY

n\k	3	4	5	6	7	10	11	12	13	15	21	22	26	27	31	40	63
4	L4 2																
8							L8 2										
9			L9 3														
12								L12 2									
16				L16b 4								L16 2					
18							L18 3,6										
25					L25 5												
27								L27 3,2									
32						L32b 4,2								L32 2			
36										L36 3		L36b 3,2					
50								L50 5,2									
54												L54 3,2					
64									L64b 2						L64 4		
81														L81 3			

廖 力

2019年10月21日

黑盒测试方法——正交数组测试

■ 选择正交表

■ 因子的水平数不同？

采用包含和组合的方法选择合适正交表

例：在4个因子中：1因子4水平，3因子3水平

选 $L_{16}(4^5)$

取最大水平数，因子数包含因子总数的正交表

- 选取了正交表后，删除正交表中多余的因子列，原则是删除靠后的列。



廖力

软件

Software Testi

Taguchi Designs

NIVERSITY

n\k	3	4	5	6	7	10	11	12	13	15	21	22	26	27	31	40	63
4	L4 2																
8							L8 2										
9			L9 3														
12								L12 2									
16				L16b 4								L16 2					
18							L18 3,6										
25					L25 5												
27								L27 3,2									
32						L32b 4,2								L32 2			
36										L36 3		L36b 3,2					
50								L50 5,2									
54												L54 3,2					
64									L64b 2						L64 4		
81														L81 3			

廖 力

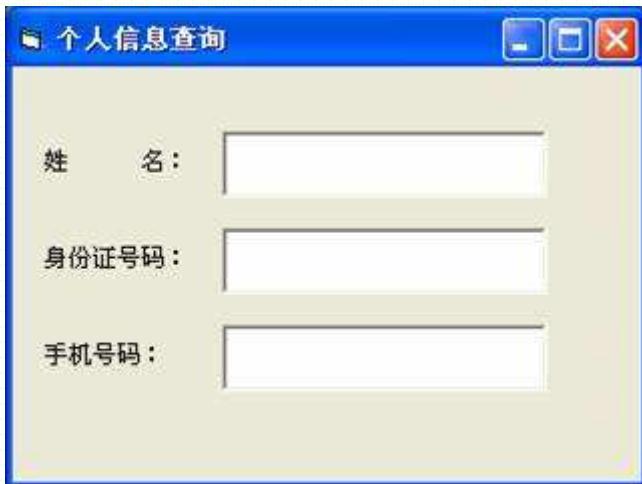
2019年10月21日

黑盒测试方法——正交数组测试

■ 正交数组测试练习1

某软件查询功能的界面如下。

查询时可根据**姓名**、**身份证号**和**手机号码**



廖 力

黑盒测试方法——正交数组测试

■ 正交数组测试练习1

因子数和水平数

三个因子：

姓名、身份证号、手机号码

每个因子都两个水平：

姓名： 填 不填

身份证号： 填 不填

手机号码： 填 不填



廖力

软件

Software Testi

Taguchi Designs

NIVERSITY

n\k	3	4	5	6	7	10	11	12	13	15	21	22	26	27	31	40	63
4	L4 2																
8							L8 2										
9			L9 3														
12								L12 2									
16				L16b 4								L16 2					
18							L18 3,6										
25					L25 5												
27								L27 3,2									
32						L32b 4,2								L32 2			
36										L36 3		L36b 3,2					
50								L50 5,2									
54												L54 3,2					
64									L64b 2						L64 4		
81														L81 3			

廖 力

2019年10月21日

黑盒测试方法——正交数组测试

■ 正交数组测试练习1

选择正交表

因子数=3

因子的水平数=2

行数取最小的正交表

结果： $L_4(2^3)$



廖 力

黑盒测试方法——正交数组测试

■ 正交数组测试练习1

变量映射

		列号		
		1	2	3
行号	1	0	0	0
	2	0	1	1
	3	1	0	1
	4	1	1	0



		列号		
		姓名	身份证号	手机号码
行号	1	填	填	填
	2	填	不填	不填
	3	不填	填	不填
	4	不填	不填	填



廖力

黑盒测试方法——正交数组测试

■ 正交数组测试练习1

测试用例

- | | | |
|---------|--------|-------|
| 1. 填姓名 | 填身份证号 | 填手机号 |
| 2. 填姓名 | 不填身份证号 | 不填手机号 |
| 3. 不填姓名 | 填身份证号 | 不填手机号 |
| 4. 不填姓名 | 不填身份证号 | 填手机号 |

增补测试用例

- | | | |
|---------|--------|-------|
| 5. 不填姓名 | 不填身份证号 | 不填手机号 |
|---------|--------|-------|

测试用例从8降为5



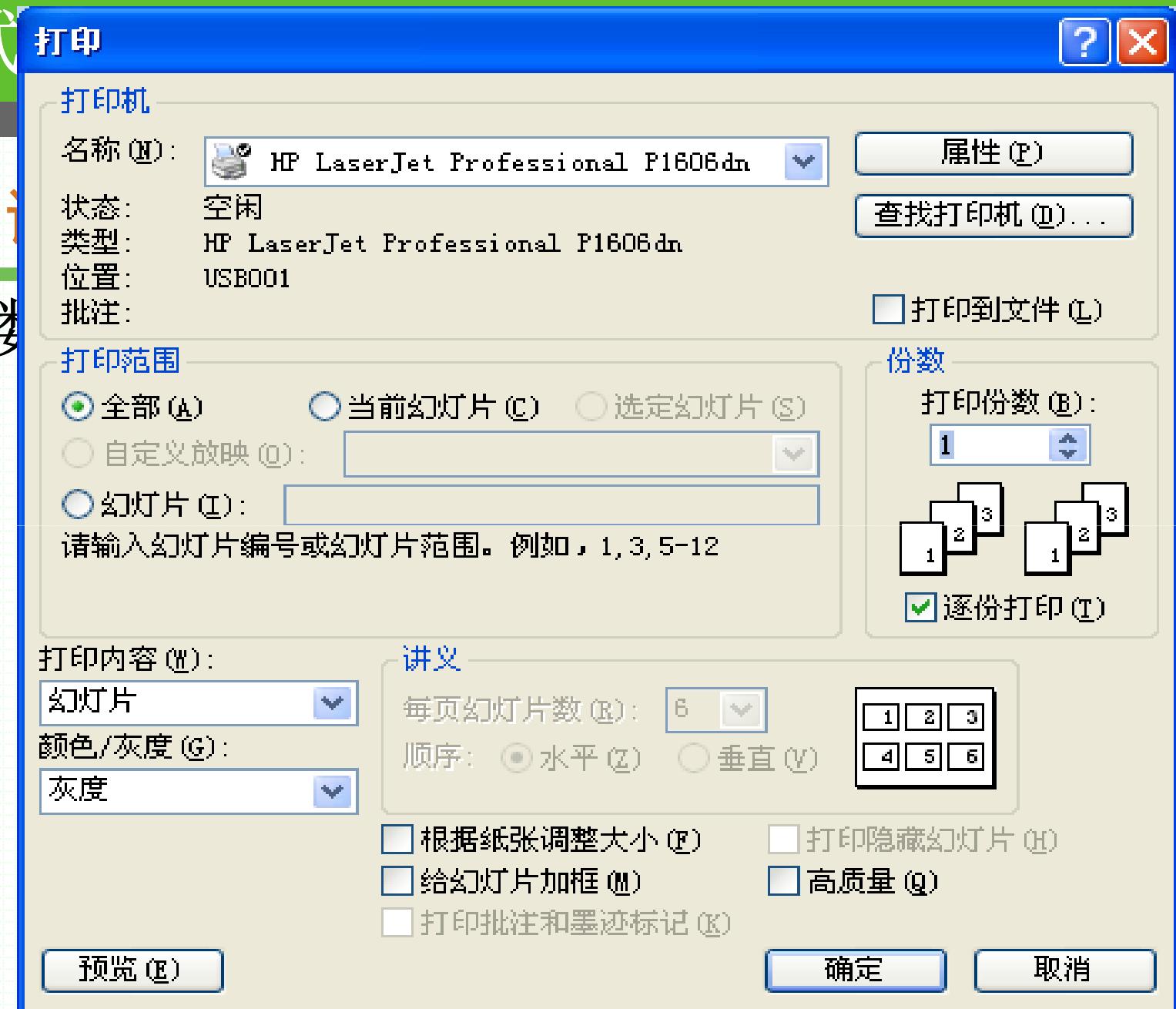
软件测试

Software Testing

黑盒测试

正交表

PowerPoint
打印功能：
**范围、内容、
颜色和效果
(仅看加框
与否)**



黑盒测试方法——正交数组测试

■ 正交数组测试练习2

因子数和水平数

	A打印范围	B打印内容	C打印颜色	D打印效果
0	全部	幻灯片	彩色	幻灯片加框
1	当前幻灯片	讲义	灰度	幻灯片不加框
2	给定范围	备注页	黑白	
3		大纲视图		



廖力

黑盒测试方法——正交数组测试

■ 正交数组测试练习2

因子数和水平数

	A打印范围	B打印内容	C打印颜色	D打印效果
0	A1	B1	C1	D1
1	A2	B2	C2	D2
2	A3	B3	C3	
3		B4		



廖力

软件

Software Testi

Taguchi Designs

NIVERSITY

n\k	3	4	5	6	7	10	11	12	13	15	21	22	26	27	31	40	63
4	L4 2																
8							L8 2										
9			L9 3														
12								L12 2									
16				L16b 4								L16 2					
18							L18 3,6										
25					L25 5												
27								L27 3,2									
32						L32b 4,2								L32 2			
36										L36 3		L36b 3,2					
50								L50 5,2									
54												L54 3,2					
64									L64b 2						L64 4		
81														L81 3			

廖 力

2019年10月21日

黑盒测试方法——正交数组测试

■ 正交数组测试练习2

选择正交表

因子数 ≥ 4

至少4个因子的水平数 ≥ 2

行数取最小的正交表

结果： $L_{16}(4^5)$



廖力

软件测试

Software Testing

黑盒测试

■ 正交数组

$L_{16}(4^5)$

	1	2	3	4	5
1	0	0	0	0	0
2	0	1	1	1	1
3	0	2	2	2	2
4	0	3	3	3	3
5	1	0	1	2	3
6	1	1	0	3	2
7	1	2	3	0	1
8	1	3	2	1	0
9	2	0	2	3	1
10	2	1	3	2	0
11	2	2	0	1	3
12	2	3	1	0	2
13	3	0	3	1	2
14	3	1	2	0	3
15	3	2	1	3	0
16	3	3	0	2	1

UNIVERSITY



廖力

黑盒测试方法——正交数组测试

—表示可取任何值

■ 正交数组测试练习2

变量映射

第5列
去除

	1	2	3	4	5
1	0	0	0	0	0
2	0	1	1	1	1
3	0	2	2	2	2
4	0	3	3	3	3
5	1	0	1	2	3
6	1	1	0	3	2
7	1	2	3	0	1
8	1	3	2	1	0
9	2	0	2	3	1
10	2	1	3	2	0
11	2	2	0	1	3
12	2	3	1	0	2
13	3	0	3	1	2
14	3	1	2	0	3
15	3	2	1	3	0
16	3	3	0	2	1

	1	2	3	4	5
1	A1	B1	C1	D1	0
2	A1	B2	C2	D2	1
3	A1	B3	C3	---	2
4	A1	B4	---	---	3
5	A2	B1	C2	---	3
6	A2	B2	C1	---	2
7	A2	B3	---	D1	1
8	A2	B4	C3	D2	0
9	A3	B1	C3	---	1
10	A3	B2	---	---	0
11	A3	B3	C1	D2	3
12	A3	B4	C2	D1	2
13	---	B1	---	D2	2
14	---	B2	C3	D1	3
15	---	B3	C2	---	0
16	---	B4	C1	---	1



黑盒测试方法——正交数组测试

	1	2	3	4	5
1	A1	B1	C1	D1	0
2	A1	B2	C2	D2	1
3	A1	B3	C3	---	2
4	A1	B4	---	---	3
5	A2	B1	C2	---	3
6	A2	B2	C1	---	2
7	A2	B3	---	D1	1
8	A2	B4	C3	D2	0
9	A3	B1	C3	---	1
10	A3	B2	---	---	0
11	A3	B3	C1	D2	3
12	A3	B4	C2	D1	2
13	---	B1	---	D2	2
14	---	B2	C3	D1	3
15	---	B3	C2	---	0
16	---	B4	C1	---	1

测试用例从72降为16



廖力

2019年10月21日

黑盒测试方法——正交数组测试

■ 正交数组测试练习3

Word字体编辑功能：

Bold、Italic、Underline和Strikethrough

	Bold B	Italic <i>I</i>	Underline <u>U</u>	Strikethrough A
Bold B	Microsoft	<i>Microsoft</i>	<u>Microsoft</u>	Microsoft
Italic <i>I</i>	<i>Microsoft</i>	<i>Microsoft</i>	<u>Microsoft</u>	<i>Microsoft</i>
Underline <u>U</u>	<u>Microsoft</u>	<u>Microsoft</u>	<u>Microsoft</u>	<u>Microsoft</u>
Strikethrough A	Microsoft	Microsoft	Microsoft	Microsoft



廖力

黑盒测试方法——正交数组测试

■ 正交数组测试练习3

因子数和水平数

Bold、Italic、Underline和**Strikethrough**

	A-Bold	B-Italic	C-Underline	D-Strikethrough
0	选中	选中	选中	选中
1	不选	不选	不选	不选

	A-Bold	B-Italic	C-Underline	D-Strikethrough
0	A1	B1	C1	D1
1	A2	B2	C2	D2



廖力

软件

Software Testi

Taguchi Designs

NIVERSITY

n\k	3	4	5	6	7	10	11	12	13	15	21	22	26	27	31	40	63
4	L4 2																
8							L8 2										
9			L9 3														
12								L12 2									
16				L16b 4								L16 2					
18							L18 3,6										
25					L25 5												
27								L27 3,2									
32						L32b 4,2								L32 2			
36										L36 3		L36b 3,2					
50								L50 5,2									
54												L54 3,2					
64									L64b 2						L64 4		
81														L81 3			

廖 力

2019年10月21日

黑盒测试方法——正交数组测试

■ 正交数组测试练习3

选择正交表

因子数 ≥ 4

因子的水平数 ≥ 2

行数取最小的正交表

$L_8(2^7)$ $L_9(3^4)$ $L_{12}(2^{11})$

结果: $L_8(2^7)$



廖力

黑盒测试方法——正交数组测试

■ 正交数组测试练习3

选择正交表

$L_8(2^7)$

	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1
2	1	1	1	2	2	2	2
3	1	2	2	1	1	2	2
4	1	2	2	2	2	1	1
5	2	1	2	1	2	1	2
6	2	1	2	2	1	2	1
7	2	2	1	1	2	2	1
8	2	2	1	2	1	1	2



廖力

黑盒测试方法——正交数组测试

■ 正交数组测试练习3

变量映射

	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1
2	1	1	1	2	2	2	2
3	1	2	2	1	1	2	2
4	1	2	2	2	1	1	
5	2	1	2	1	2	1	2
6	2	1	2	2	1	2	1
7	2	2	1	1	2	2	1
8	2	2	1	2	1	1	2

第5,6,7
列去除

	1	2	3	4	5	6	7
	A1	B1	C1	D1	1	1	1
	A1	B1	C1	D2	2	2	2
	A1	B2	C2	D1	1	2	2
	A1	B2	C2	D2	2	1	1
	A2	B1	C2	D1	2	1	2
6	A2	B1	C2	D2	1	2	1
7	A2	B2	C1	D1	2	2	1
8	A2	B2	C1	D2	1	1	2



廖 力

黑盒测试方法——正交数组测试

■ 正交数组测试练习3

测试用例

	1	2	3	4
1	A1	B1	C1	D1
2	A1	B1	C1	D2
3	A1	B2	C2	D1
4	A1	B2	C2	D2
5	A2	B1	C2	D1
6	A2	B1	C2	D2
7	A2	B2	C1	D1
8	A2	B2	C1	D2
9	A2	B2	C2	D2

测试用例从16降为8

进一步增加遗漏的重要测试用例，如9



廖力

黑盒测试方法总结

特点	被测软件内部逻辑未知的测试方法总称
目的	功能的正确和遗漏；界面错误；数据访问；性能；初始化和终止
技术	基于需求的测试；正面测试和负面测试； 边界值分析；因果分析法； 决策表；等价类划分； 基于图/状态的测试；正交数组测试
插桩	实现白盒测试和黑盒测试的支持技术



廖力

灰盒测试



廖 力

东南大学 软件学院 College of Software Engineering

2019年10月21日

灰盒测试简介

■ 定义

定义1：基于程序内部细节有限认知的软件调试方法。

定义2：

*Test designed based on the knowledge of algorithm, internal states, architectures, or other **high -level descriptions of the program behavior.** [Doug Hoffman]*

测试者可能知道系统组件之间的相互作用，但缺乏对内部程序功能和运作的详细了解。

对内部细节的有限了解有利于设计更好的测试用例。



廖 力

灰盒测试简介

■ 白盒-黑盒-灰盒

	白盒测试	黑盒测试	灰盒测试
代码依赖	需要源代码	不需要源代码	需要了解实现的有限细节
实施时机	单元测试	单元测试后	集成测试前期
实施者	开发人员	测试人员	测试人员
关注对象	内部实现	功能需求	二者兼有
测试方法	代码审查和代码分析	设计并执行测试用例	编写测试代码
是否必须	不能被取代	不能被取代	可被黑盒测试取代



腐力

三 黑盒测试工具



1. 主流的黑盒测试工具有哪些？
2. 测试工具如何使用？



廖力

功能测试工具

■ 特点

黑盒测试工具；

软件测试工具中最活跃的类型；

多用于系统测试、确认测试和回归测试；

以自动化测试工具为主



廖 力

功能测试工具

■ 功能测试工具的功能

1. 录制和回放

录制: 测试输入的自动化

回放: 重新执行测试用例

2. 检验

检验: 设置检测点, 对用例执行的正确性进行检查

3. 可编程

编程: 脚本更加灵活、功能更强大



廖力

功能测试工具

■ 自动化测试工具原理

（以基于GUI的自动化测试工具为例）：

运行被测软件的同时，捕获过程中的键盘、鼠标操作，生成脚本文件，这个脚本文件可以进行修改和回放。



廖 力

功能测试工具

■ 自动化测试工具示例

- 开源测试工具Selenium的使用
- <http://seleniumhq.org/download/>
- Selenium ide
- Selenium IDE Dec 6, 2010 1.0.10

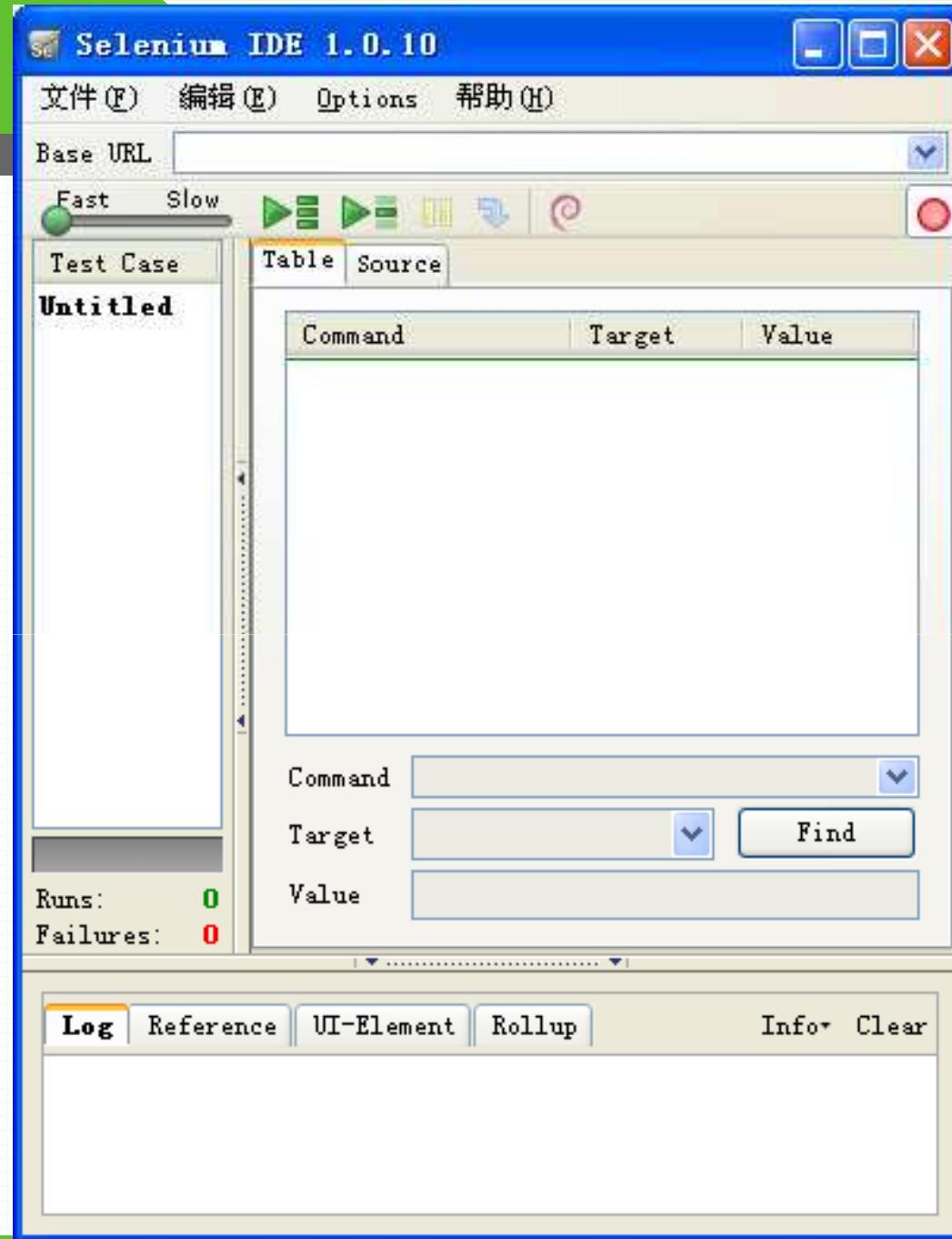


廖 力



软件测试

Software Testing



ST UNIVERSITY



廖 力

东南大学 软件学院 College of Software Engineering

2019年10月21日

软件测试

SOUTHEAST UNIVERSITY

Software Testing

百度搜索_自动化测试 - Mozilla Firefox

文件(F) 编辑(E) 查看(V) 历史(S) 书签(B) 工具(T) 帮助(H)

http://www.baidu.com/s?wd=%D7%D4%B6%AF%BB%AF%B2%E2%CAX%D4

访问最多 新手上路 最新头条

百度搜索_自动化测试

新闻 网页 贴吧 知道 MP3 图片 视频 地图 更多▼

Baidu 百度

自动化测试

百度一下

把百度设为主页

百度一下，找到相关网页约2,310,000篇，用时0.004秒

[自动化测试 百度百科](#)

一般是指软件**测试**的**自动化**，**软件测试**就是在预设条件下运行系统或应用程序，评估运行结果，预先条件应包括正常条件和异常条件。共8次编辑

[概述](#) [前提条件](#) [适用诫](#) - 方案选型六大原则

baike.baidu.com/view/1303916.htm 2010-5-12

[自动化测试框架 百度文库](#)

推广链接

[达鑫-自动化测试专家](#)
为您量身定制**自动化测试**系统，
高性价比,稳定可靠,优质服务.
www.daystarchn.com

自动化测试首选**百度**质量保证



廖 力

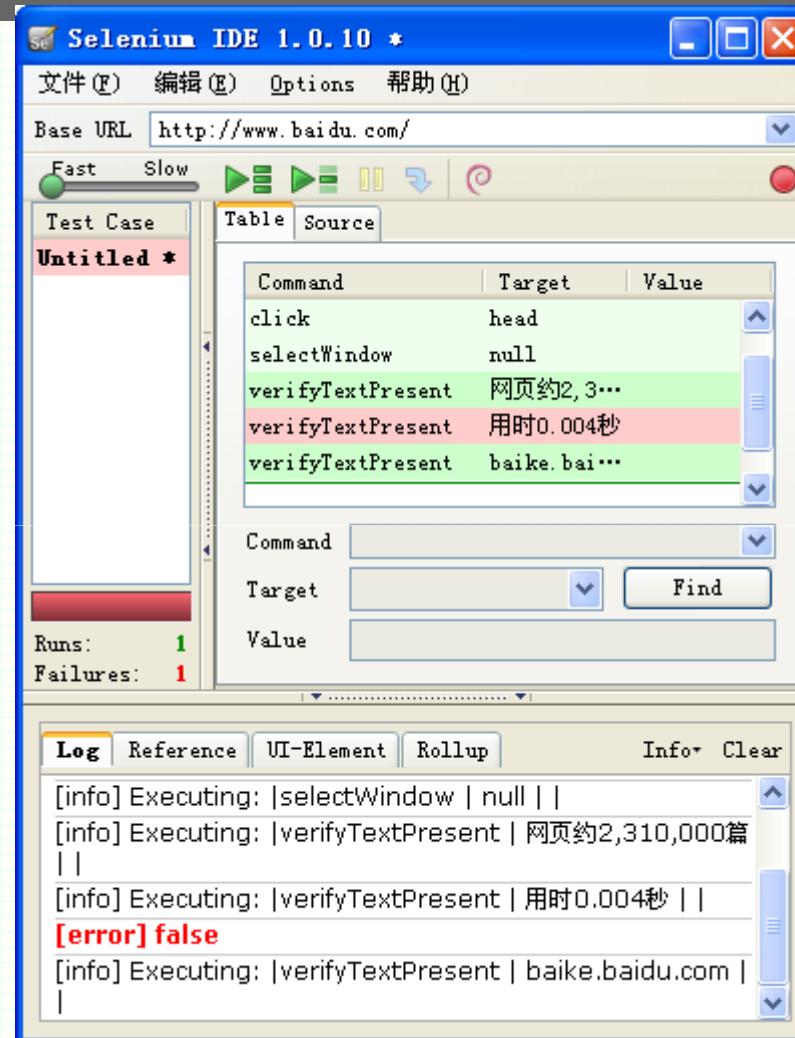
东南大学 软件学院 College of Software Engineering

2019年10月21日

软件测试

SOUTHEAST UNIVERSITY

Software Testing



廖 力

东南大学 软件学院 College of Software Engineering

2019年10月21日

功能测试工具介绍

■ 主流功能测试工具

1. Quick Test Professional (QTP)
2. Rational Robot
3. SilkTest
4. AutoRunner



廖力

功能测试工具介绍

■ Quick Test Professional

功能：

创建功能和回归测试；

自动捕获、验证和重放用户的交互行为；

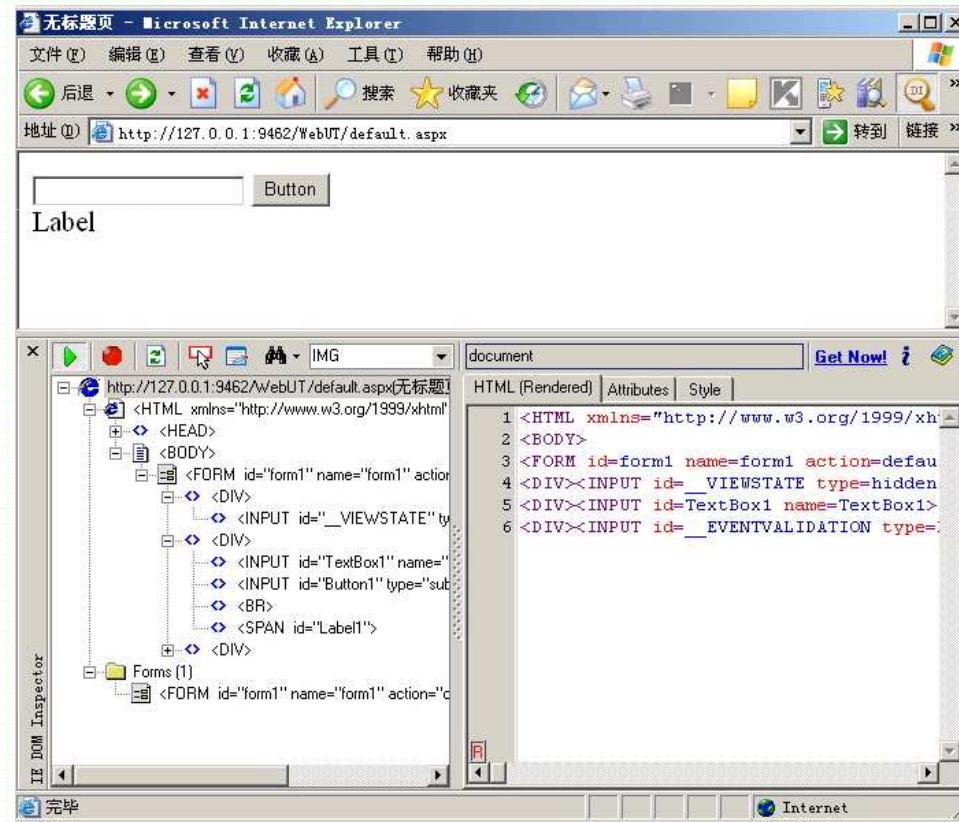
为每一个重要软件应用和环境提供功能和回归测试自动化的行业最佳解决方案。



廖 力

功能测试工具介绍

■ Quick Test Professional



廖 力

功能测试工具介绍

■ Robot

功能：一种对环境的多功能的、回归和配置测试工具
特点：

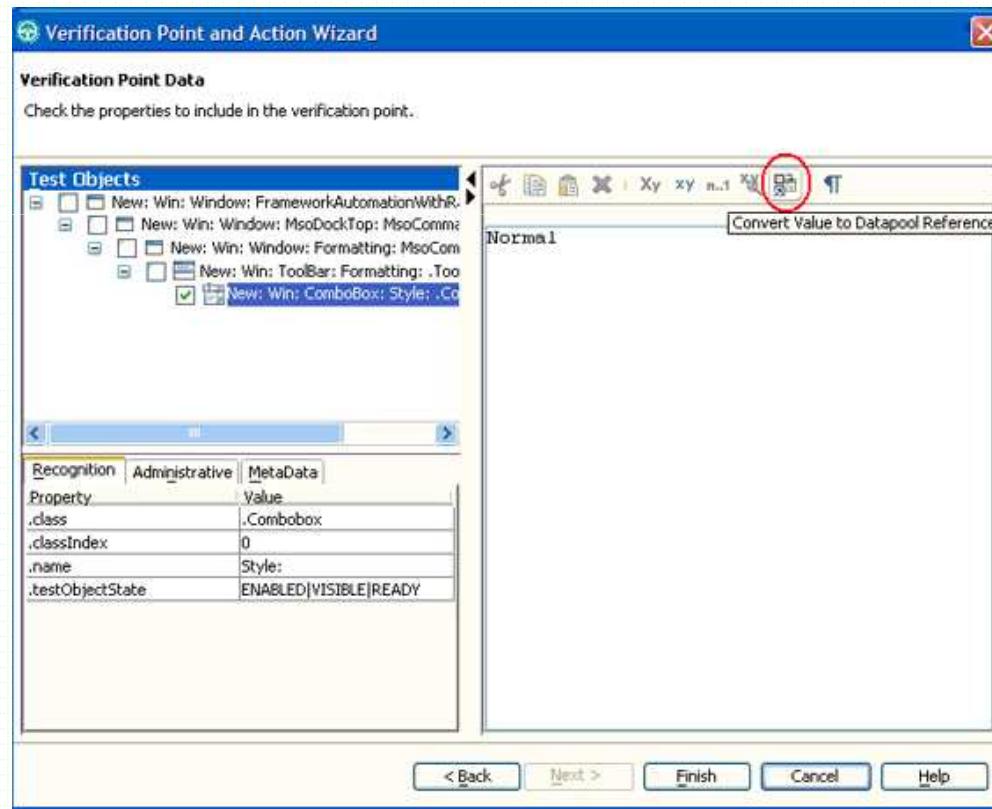
1. 测试深度：测试由应用的 UI 扩展到应用组件对象的上百种属性
2. 可测试众多类型的应用：HTML 和 DHTML、JavaTM、VS.NET、Microsoft Visual Basic 和 Visual C++ 等
3. 测试自定义控件和对象：提供菜单、列表、文字数字式字符、位图等许多对象的测试用例
4. 提供集成化编程环境：QABasic 脚本
5. 帮助您快速分析问题：友好的数据整合



廖 力

功能测试工具介绍

■ Robot



功能测试工具介绍

■ SilkTest

功能：是面向**Web**应用、**Java**应用和传统的**C/S**应用，进行自动化的功能测试和回归测试的工具

特点：

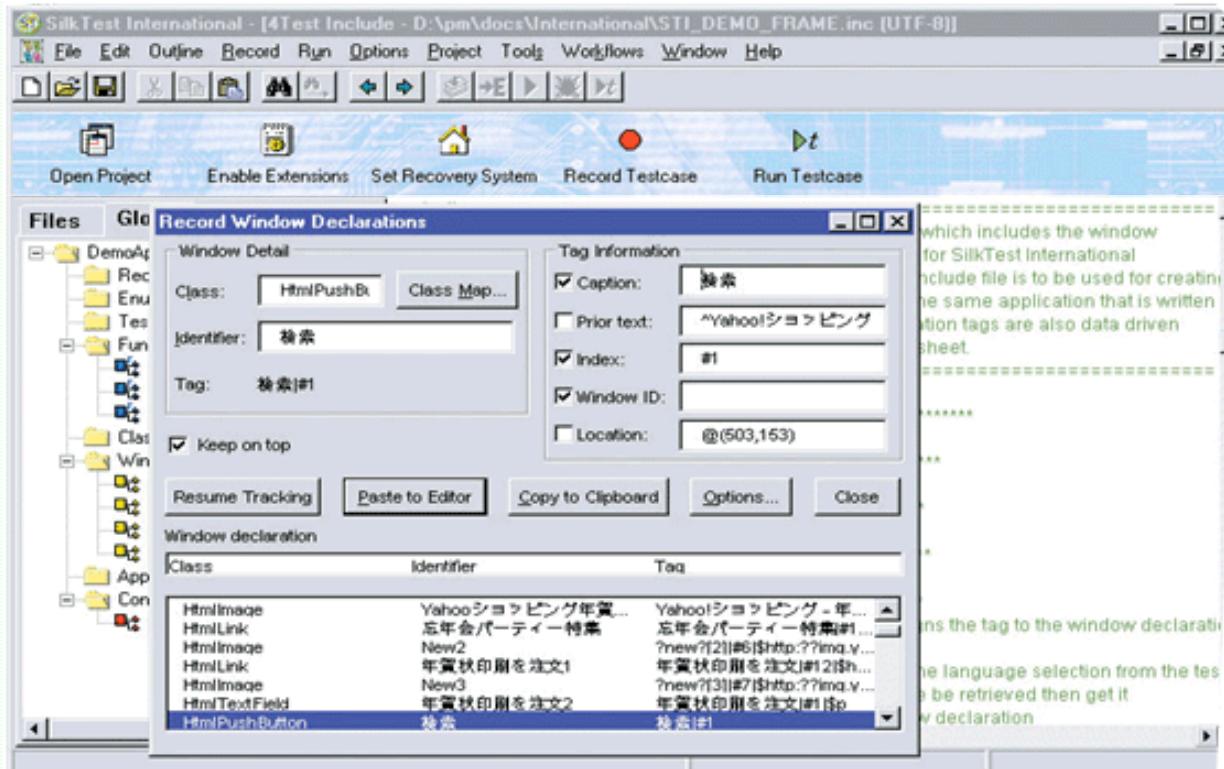
提供了用于测试的创建和定制的工作流设置、测试计划和管理、直接的数据库访问及校验等功能，使用户能够高效率地进行软件自动化测试



廖 力

功能测试工具介绍

SilkTest



功能测试工具介绍

■ AutoRunner

功能：

1. Web测试——支持各种B/S应用和网站
2. .NET测试——对.NET类型的应用软件进行功能测试

特点：

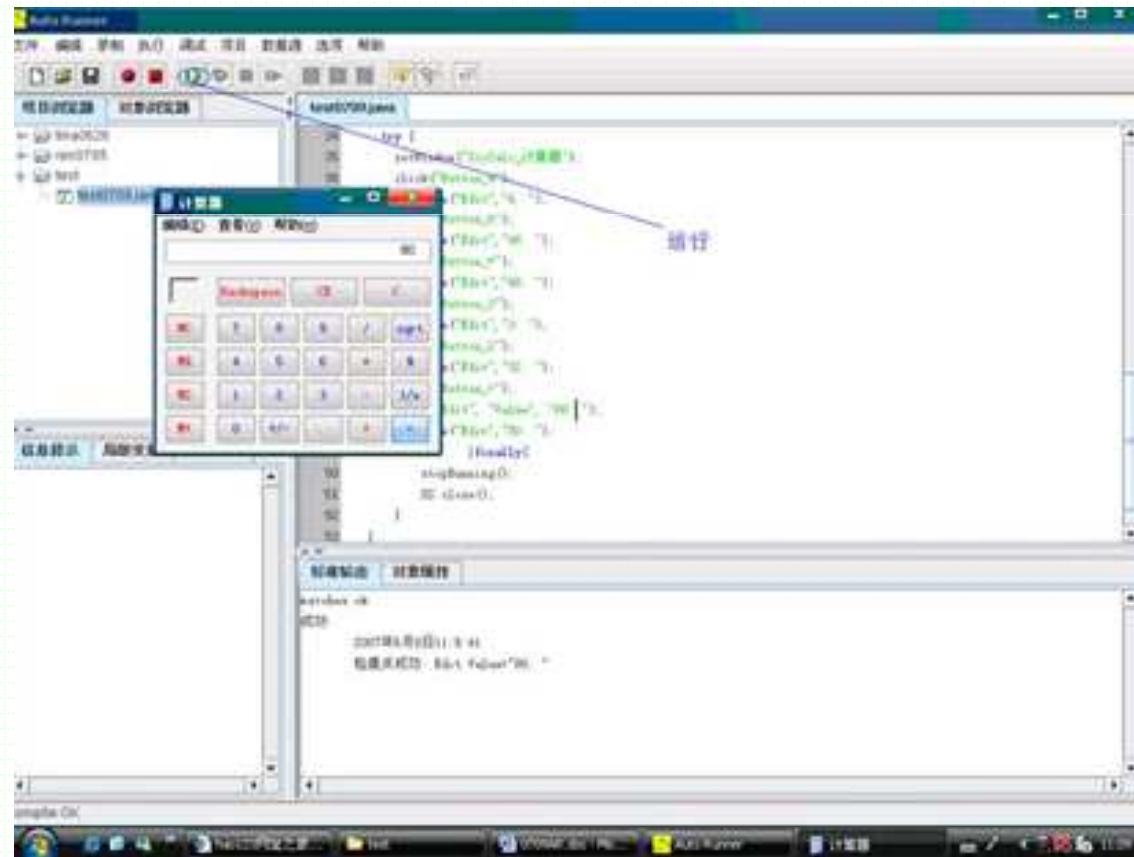
AutoRunner是黑盒测试工具，可以用来完成功能测试、回归测试、每日构建测试与自动回归测试等工作。是具有脚本语言的、提供完善的针对脚本跟踪和调试功能的、支持IE测试和Windows 测试的自动化测试工具。



廖 力

功能测试工具介绍

■ AutoRunner



廖 力

黑盒测试工具介绍

Parasoft C++ Test演示



廖 力

2019年10月21日

参考文献

1. 郁莲, 软件测试方法与实践, p45-p60, 清华大学出版社
2. 宫云战 主编, 软件测试教程, p19-p48, 机械工业出版社
3. Andreas Spillner, Tilo Linz, Hans Schaefer (刘琴 等 译), 软件测试基础教程, p81-p112, 人民邮电出版社
4. Srinivasan Desikan, Gopalaswamy Ramesh, (韩柯, 李娜 译), 软件测试: 原理与实践, p47-p66, 机械工业出版社
5. 路晓丽, 葛玮, 龚晓庆, 软件测试技术, p35-p50, 机械工业出版社
6. Grey Box Testing,
<http://www.geocities.com/xtremetesting/GreyBox.html>
7. <http://bbs.zbitedu.com/archiver/tid-23088.html>
8. 朱少民, 软件测试, 人民邮电出版社
9. 微软的软件测试之道



廖 力

3.4 补充确定的关联边界条件

边界条件	月	日	年
15	1	19	2038
16	10	5	1582
17	10	14	1582
18	9	3	1752
19	9	13	1752

$m=19$

总的测试用例数： $19 \times 3 = 57$